

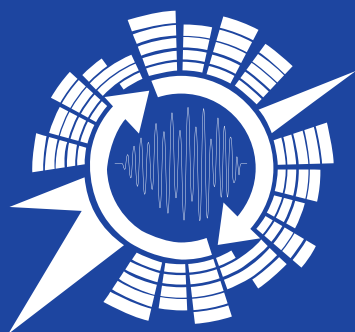
Alexey Karpov
Gábor Gosztolya (Eds.)

LNAI 16188

Speech and Computer

27th International Conference, SPECOM 2025
Szeged, Hungary, October 13–15, 2025
Proceedings, Part II

2
Part II



 Springer

Lecture Notes in Computer Science

Lecture Notes in Artificial Intelligence

16188

Founding Editor

Jörg Siekmann

Series Editors

Randy Goebel, *University of Alberta, Edmonton, Canada*

Wolfgang Wahlster, *DFKI, Berlin, Germany*

Zhi-Hua Zhou, *Nanjing University, Nanjing, China*

The series Lecture Notes in Artificial Intelligence (LNAI) was established in 1988 as a topical subseries of LNCS devoted to artificial intelligence.


The series publishes state-of-the-art research results at a high level. As with the LNCS mother series, the mission of the series is to serve the international R & D community by providing an invaluable service, mainly focused on the publication of conference and workshop proceedings and postproceedings.

Alexey Karpov · Gábor Gosztolya
Editors

Speech and Computer

27th International Conference, SPECOM 2025
Szeged, Hungary, October 13–15, 2025
Proceedings, Part II

Editors

Alexey Karpov 
St. Petersburg Federal Research Center
of the Russian Academy of Sciences
St. Petersburg, Russia

Gábor Gosztolya 
University of Szeged
Szeged, Hungary

ISSN 0302-9743

ISSN 1611-3349 (electronic)

Lecture Notes in Artificial Intelligence

ISBN 978-3-032-07958-9

ISBN 978-3-032-07959-6 (eBook)

<https://doi.org/10.1007/978-3-032-07959-6>

LNCS Sublibrary: SL7 – Artificial Intelligence

© The Editor(s) (if applicable) and The Author(s), under exclusive license
to Springer Nature Switzerland AG 2026

This work is subject to copyright. All rights are solely and exclusively licensed by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

If disposing of this product, please recycle the paper.

SPECOM 2025 Preface

SPECOM is a conference with a long tradition that attracts researchers in the area of speech technology, including automatic speech recognition and understanding, text-to-speech synthesis, and speaker and language recognition, as well as related domains like digital speech processing, natural language processing, text analysis, computational paralinguistics, multi-modal speech, and data processing or human-computer interaction. The SPECOM conference is an ideal platform for know-how exchange – especially for experts working on inflective or agglutinative spoken languages – including both under-resourced and well-resourced ones.

The International Conference on Speech and Computer (SPECOM) has become a regular event since the first SPECOM was held in St. Petersburg, Russia, in October 1996. The SPECOM conference series was established exactly 29 years ago by the St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences (SPIIRAS).

In its long history, the SPECOM conference has been organized alternately by the St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS)/SPIIRAS and by the Moscow State Linguistic University (MSLU) in their home towns. Furthermore, in 1997 it was organized by the Cluj-Napoca subsidiary of the Research Institute for Computer Technique (Romania), in 2005 and 2015 by the University of Patras (in Patras and Athens, Greece), in 2011 by the Kazan Federal University (in Kazan, Russia), in 2013 by the University of West Bohemia (in Pilsen, Czech Republic), in 2014 by the University of Novi Sad (in Novi Sad, Serbia), in 2016 by the Budapest University of Technology and Economics (in Budapest, Hungary), in 2017 by the University of Hertfordshire (in Hatfield, UK), in 2018 by the Leipzig University of Telecommunications (in Leipzig, Germany), in 2019 by the Bogaziçi University (in Istanbul, Turkey), in 2020 and 2021 by SPC RAS/SPIIRAS (fully online), in 2022 by the KIIT (in Gurugram, New Delhi, India), in 2023 by the IIT/IIIT Dharwad (in Hubli-Dharwad, Karnataka, India), and in 2024 by the University of Novi Sad (in Belgrade, Serbia).

SPECOM 2025 (<https://specom.inf.u-szeged.hu>) was the 27th event in the conference series, and the second time SPECOM was in Hungary. SPECOM 2025 was organized by the Institute of Informatics of the University of Szeged. The conference was held from 13th till 14th October 2025, in a hybrid format, mostly in-person at the Novotel Hotel Szeged and online via video conferencing. SPECOM 2025 was also supported by the International Speech Communication Association (ISCA).

During SPECOM 2025, two keynote lectures were given by Éva Székely (Department of Speech, Music and Hearing, KTH Royal Institute of Technology, Stockholm, Sweden) on “From Conversation to Conversational: Speech Synthesis and the Communicative Power of the Human Voice”, as well as by Heysem Kaya (Social and Affective Computing Group, Department of Information and Computing Sciences, Utrecht University, Utrecht, the Netherlands) on “Towards Responsible Multimodal Modeling for Mental Healthcare”.

The two volumes of the SPECOM 2025 proceedings contain a collection of submitted papers presented at SPECOM 2025, which were thoroughly reviewed by members of the Program Committee and additional reviewers consisting of over 70 experts in the conference topic areas. In total, 47 regular full papers out of 77 submissions made via the EasyChair electronic system were carefully selected by the SPECOM 2025 Program Committee members for oral and poster presentations at the conference, as well as for inclusion in the SPECOM 2025 proceedings. Each valid submission was reviewed in a single-blind manner by at least three members of the Program Committee. Theoretical and more general contributions were presented in common plenary sessions. Problem-oriented sessions as well as panel discussions brought together specialists in niche problem areas with the aim of exchanging knowledge and skills resulting from research projects of all kinds.

We would like to express our gratitude to all authors for providing their papers on time, to the members of the SPECOM 2025 Program Committee for their careful reviews and paper selection, and to the editors and correctors for their hard work in preparing the conference proceedings. Special thanks are due to the members of the SPECOM 2025 Organizing Committee for their tireless effort and enthusiasm during the conference organization. We are also grateful to the Institute of Informatics of the University of Szeged for organizing and hosting the 27th International Conference on Speech and Computer SPECOM 2025 in the city of Szeged.

October 2025

Alexey Karpov
Gábor Gosztolya

Organization

General Chairs

Gábor Gosztolya

Alexey Karpov

University of Szeged, Hungary

St. Petersburg Federal Research Center of the
Russian Academy of Sciences, Russia

Program Committee

Alexey Karpov (Chair)

Gábor Gosztolya (Chair)

Mohammed Al-Radhi

Jahangir Alam

Alexandr Axyonov

Árpád Berta

Milana Bojanić

Vladimir Chuchupal

St. Petersburg Federal Research Center of the
Russian Academy of Sciences, Russia

University of Szeged, Hungary

Budapest University of Technology and
Economics, Hungary

Computer Research Institute of Montreal
(CRIM), Canada

St. Petersburg Federal Research Center of the
Russian Academy of Sciences, Russia

University of Szeged, Hungary

University of Novi Sad, Serbia

Federal Research Center “Computer Science and
Control” of Russian Academy of Sciences,
Russia

Andrea Corradini

Vlado Delić

Anna Esposito

MCI Innsbruck, Austria

University of Novi Sad, Serbia

Università degli Studi della Campania
“L. Vanvitelli”, Italy

Yannick Estève

Vera Evdokimova

Olga Frolova

Philip N. Garner

Branislav Gerazov

Avignon University, France

St. Petersburg State University, Russia

St. Petersburg State University, Russia

Idiap Research Institute, Switzerland

Ss. Cyril and Methodius University, North
Macedonia

Ivan Gruber

Tamás Grósz

Rüdiger Hoffmann

Denis Ivanko

University of West Bohemia, Czech Republic

Aalto University, Finland

TU Dresden, Germany

St. Petersburg Federal Research Center of the
Russian Academy of Sciences, Russia

Nikša Jakovljević	University of Novi Sad, Serbia
Ildar Kagirov	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Heysem Kaya	Utrecht University, the Netherlands
Maria Khokhlova	St. Petersburg State University, Russia
Irina Kipyatkova	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Olesia Koroteeva	ITMO University, Russia
Evgeny Kostyuchenko	Tomsk State University of Control Systems and Radioelectronics, Russia
György Kovács	Luleå University of Technology, Sweden
Ivan Kraljevski	Fraunhofer IKTS, Germany
Yanxiong Li	South China University of Technology, China
Natalia Loukachevitch	Lomonosov Moscow State University, Russia
Elena Lyakso	St. Petersburg State University, Russia
Ilya Makarov	Artificial Intelligence Research Institute, Russia
Maxim Markitantov	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Konstantin Markov	University of Aizu, Japan
Yuri Matveev	ITMO University, Russia
Peter Mihajlik	Budapest University of Technology and Economics, Hungary
Nikolay Mikhaylovskiy	Tomsk State University, Russia
Rathish Mohan	Sequelae Inc., India
Bernd Möbius	Saarland University, Germany
Oliver Niebuhr	University of Southern Denmark, Denmark
Dariya Novokhrestova	Tomsk State University of Control Systems and Radioelectronics, Russia
Sergey Novoselov	STC-innovations Ltd., Russia
Géza Németh	Budapest University of Technology and Economics, Hungary
Nick A. Petrovsky	Belarusian State University of Informatics and Radioelectronics, Belarus
Branislav Popović	University of Novi Sad, Serbia
Vsevolod Potapov	Lomonosov Moscow State University, Russia
Rodmonga Potapova	Moscow State Linguistic University, Russia
Sergey Rybin	ITMO University, Russia
Dmitry Ryumin	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Elena Ryumina	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Albert Ali Salah	Utrecht University, the Netherlands
Milan Sečujski	University of Novi Sad, Serbia

Tatiana Sherstinova	HSE University, Russia
Nickolay Shmyrev	Alpha Cephei Inc., Russia
Nikola Simić	University of Novi Sad, Serbia
Pavel Skrelin	St. Petersburg State University, Russia
Tatiana Sokoreva	Moscow State Linguistic University, Russia
Claudia Soria	Istituto di Linguistica Computazionale CNR, Italy
Victor Sorokin	Institute for Information Transmission Problems, Russia
Siniša Suzić	University of Novi Sad, Serbia
Dávid Sztahó	Budapest University of Technology and Economics, Hungary
Zsolt Szántó	University of Szeged, Hungary
Ivan Tashev	Microsoft, USA
Laszlo Toth	University of Szeged, Hungary
Jan Trmal	Johns Hopkins University, USA
Liliya Tsirulnik	Stenograph LLC, USA
Maxim Vashkevich	Belarusian State University of Informatics and Radioelectronics, Belarus
Alena Velichko	St. Petersburg Federal Research Center of the Russian Academy of Sciences, Russia
Veronika Vincze	Hungarian Academy of Sciences, Hungary
Zeynep Yucel	Okayama University, Japan
Csaba Zainkó	Budapest University of Technology and Economics, Hungary
Jerneja Zganec Gros	Alpineon Research and Development Ltd., Slovenia

Additional Reviewers

Nikolay Bobrov
Mikhail Dolgushin
Jovan Galić
Ibrahim Ibrahimov
Danila Mamontov
Elena Shamina
Vuk Stanojev

Organizing Committee


Gábor Gosztolya (Chair)	University of Szeged, Hungary
Veronika Vincze	Hungarian Academy of Sciences, Hungary
Laszlo Toth	University of Szeged, Hungary
Mercedes Kiss-Vetráb	University of Szeged, Hungary
Alexey Karpov	SPC RAS, Russia
Dmitry Ryumin	SPC RAS, Russia
Irina Kipyatkova	SPC RAS, Russia
Ildar Kagiroy	SPC RAS, Russia

Keynote Speakers

Éva Székely	KTH Royal Institute of Technology, Sweden
Heysem Kaya	Utrecht University, the Netherlands

Keynotes

From Conversation to Conversational: Speech Synthesis and the Communicative Power of the Human Voice

Éva Székely 

Division of Speech, Music and Hearing, KTH Royal Institute of Technology,
Lindstedtsvägen 24, SE-114 28 Stockholm, Sweden

szekely@kth.se

<https://www.kth.se/profile/szekely>

Abstract. Deep-learning-based speech synthesis now allows us to generate voices that are not only natural-sounding but also highly realistic and expressive. This capability presents a paradox for conversational AI: it opens up new possibilities for more fluid, humanlike interaction, yet it also exposes a gap in our understanding of how such expressive features shape communication. Can synthetic speech, which poses these challenges, also help us solve them? In this talk, I explore the fundamental challenges in modelling the spontaneous phenomena that characterise spoken interaction: the timing of breaths, shifts in speech rate, laughter, hesitations, tongue clicks, creaky voice and breathy voice. In striving to make synthetic speech sound realistic, we inevitably generate communicative signals that convey stance, emotion, and identity. Modelling voice as a social signal raises important questions: How does gender presentation in synthetic speech influence perception? How do prosodic patterns affect trust, compliance, or perceived politeness? To address such questions, I will present a methodology that uses controllable conversational TTS not only as a target for optimisation but also as a research tool. By precisely manipulating prosody and vocal identity in synthetic voices, we can isolate their effects on listener judgments and experimentally test sociopragmatic hypotheses. This dual role of TTS – as both the object of improvement and the instrument of inquiry – requires us to rethink evaluation beyond mean opinion scores, towards context-driven and interaction-aware metrics. I will conclude by situating these ideas within the recent paradigm shift toward large-scale multilingual TTS models and Speech LLMs, outlining research directions that help us both understand and design for the communicative power of the human voice.

Keywords: Speech Synthesis · Speech Technology · Human Voice

Towards Responsible Multimodal Modeling for Mental Healthcare

Heysem Kaya¹  and Gizem Sogancioglu² 

¹ Department of Information and Computing Sciences, Utrecht University, Princetonplein 5,
3584 CC Utrecht, the Netherlands

h.kaya@uu.nl

² Department of Psychiatry, University Medical Center Utrecht, Heidelberglaan 100, 3584 CX
Utrecht, the Netherlands

g.sogancioglu@umcutrecht.nl

<https://www.uu.nl/staff/HKaya>

Abstract. Mood disorders, especially major depression and bipolar mania, are among the leading causes of disability worldwide. In clinical practice, the diagnosis of mood disorders is done by the medical experts via multiple observations and by means of questionnaires. This system is however subjective, costly, and cannot meet diagnostic needs given the increasing demand, risking a large population of patients with insufficient care. Increasingly in the last decade, many Artificial Intelligence (AI) and particularly Machine Learning (ML) based solutions were proposed to respond to the urgent need for objective, efficient, and effective mental healthcare decision support systems to assist and reduce the load of the medical experts. However, many of these methods lack properties for being “responsible AI”, namely, interpretability/explainability, algorithmic fairness, and privacy considerations (in both their design and final outputs), thus rendering them useless in real life, especially in the light of recent legal developments. This paper aims to provide an overview on the motivations, recent efforts, and potential future directions for responsible multimodal modeling in mental healthcare.

Keywords: Fair machine learning · Explainable AI · Mental health

Contents – Part II

Automatic Speech Recognition

In-Domain SSL Pre-training and Streaming ASR: Application to Air Traffic Control Communications	3
<i>Jarod Duret, Salima Mdhaffar, Gaëlle Laperrière, Ryan Whetten, Audrey Galametz, Catherine Kobus, Marion-Cécile Martin, Jo Oleiwan, and Yannick Estève</i>	
Evaluating the Performance of Several ASR Systems in Environmental and Industrial Noise	13
<i>Sara M. Pearsell, Oliver Niebuhr, and Samuel Schmück</i>	
Ground Truth-Free WER Prediction for ASR via Audio Quality and Model Confidence Features	29
<i>Anton Polevoi, Alexander Kragin, and Natalia Loukachevitch</i>	
Enhancing Speech Recognition Through Text-to-Speech and Voice Conversion Augmentation	45
<i>Yunus Emre Ozkose and Ali Haznedaroglu</i>	
Best Data is more Supervised Data – Even for Hungarian ASR	60
<i>Gergely Dobsinszki, Péter Mihajlik, Máté Soma Kádár, Tibor Fegyó, and Katalin Mády</i>	
Arabic ASR on the SADA Large-Scale Arabic Speech Corpus with Transformer-Based Models	70
<i>Branislav Gerazov, Marcello Politi, and Sébastien Bratières</i>	
Speech Processing for Under-Resourced Languages	
Effect of Increased Temporal Resolution on Speech Recognition for French Quebec Using Features from Speech Self-supervised Learning Models	87
<i>Vishwa Gupta and Gilles Boulianne</i>	
Modeling Intra-word Code-Switching for Karelian ASR	104
<i>Irina Kipyatkova, Kseniia Kiseleva, Mikhail Dolgushin, and Ildar Kagirov</i>	

Improving Whisper-Based Serbian ASR Using Synthetic Speech	118
<i>Vuk Stanojev, Tijana Nosek, Siniša Suzić, Darko Pekar, Vlado Delić, and Milan Sečujski</i>	

Domain Knowledge and Language Embeddings for Low-Resource Multilingual Phoneme ASR	130
<i>Anton Legchenko and Ivan Bondarenko</i>	

Whistler Identification in Whistled Spanish (Silbo): A Case Study	144
<i>Alejandro López-García, María Alfaro-Contreras, Julien Meyer, and Jose J. Valero-Mas</i>	

Digital Speech Processing

PinkVocalTransformer: Neural Acoustic-to-Articulatory Inversion Based on the Pink Trombone	161
<i>Zhiyuan Xu and Joshua Reiss</i>	

CrossMP-SENet: Transformer-Based Cross-Attention for Joint Magnitude-Phase Speech Enhancement	174
<i>Alexander Zaburdaev, Denis Ivanko, and Dmitry Ryumin</i>	

Adaptive Singing Voice Enhancement for Live Stages	189
<i>Jia-Lien Hsu and Pei-Wen Chien</i>	

Revealing the Hidden Temporal Structure of HubertSoft Embeddings Based on the Russian Phonetic Corpus	203
<i>Anastasia Ananeva, Anton Tomilov, and Marina Volkova</i>	

Natural Language Processing

Analyzing Web-Scraped and Generated Inputs for Automatic and Scalable Intent Classification	219
<i>Philine Kowol and Stefan Hillmann</i>	

Enhancing Retrieval Performance via LLM Hard-Negative Filtering	231
<i>Danil Tirskikh, Olesia Koroteeva, Yuri Matveev, Ekaterina Brovkina, and Larisa Gonchar</i>	

Sector-Wise Backpropagation for Low-Resource Text Classification in Deep Models	242
<i>José Luis Vázquez Noguera, Carlos U. Valdez, Marvin M. Agüero, Julio C. Mello, José D. Colbes, and Sebastián A. Grillo</i>	

High-Frequency Multiword Units and the Typological Distribution of Multiword Units in Spoken Russian	257
<i>Natalia V. Bogdanova-Beglarian, Olga V. Blinova, Maria V. Khokhlova, Tatiana Y. Sherstinova, and Tatiana I. Popova</i>	
Estimation of the Genre Composition of the English Subcorpus of the Google Books Ngram	271
<i>Vladimir Bochkarev, Andrey A. Achkeev, and Anna Shevlyakova</i>	
Multimodal Systems	
Ensembling Synchronisation-Based and Face-Voice Association Paradigms for Robust Active Speaker Detection in Egocentric Recordings	289
<i>Jason Clarke, Yoshihiko Gotoh, and Stefan Goetze</i>	
Phonetic and Visual Characteristics of Cognitive Load	302
<i>Vera Evdokimova and Maria Maksimova</i>	
Cognitive Humor Processing in the Russian and English Internet Meme Chatting: EEG Study	318
<i>Rodmonga Potapova, Vsevolod Potapov, Ekaterina Karimova, Diana Smolskaya, Nikolay Bobrov, Leonid Motovskikh, and Iurii Pozhilov</i>	
Saudi Sign Language Translation Using T5	331
<i>Ali Alhejab, Tomáš Železný, Lamya Alkanhal, Ivan Gruber, Yazeed Alharbi, Jakub Straka, Václav Javorek, Marek Hruží, Badriah Alkalifah, and Ahmed Ali</i>	
Author Index	345

Contents – Part I

Invited Paper

Towards Responsible Multimodal Modeling for Mental Healthcare	3
<i>Heysem Kaya and Gizem Sogancioglu</i>	

Speech Perception and Synthesis

When Voice Matters: Evidence of Gender Disparity in Positional Bias of SpeechLLMs	25
<i>Shree Harsha Bokkahalli Satish, Gustav Eje Henter, and Éva Székely</i>	
WhiSQA: Non-intrusive Speech Quality Prediction Using Whisper Encoder Features	39
<i>George Close, Kris Hong, Thomas Hain, and Stefan Goetze</i>	
Prompting the Mind: EEG-to-Text Translation with Multimodal LLMs and Semantic Control	52
<i>Mohammed Salah Al-Radhi, Sadi Mahmud Shurid, and Géza Németh</i>	
Effectiveness of Tacotron2 for Intonation Model Synthesis in Russian	67
<i>Anastasiia Sherban and Uliana Kochetkova</i>	
Enhancing Sinhala Text-to-Speech with End-to-End VITS Architecture	83
<i>Sasangi Nayanathara, Inuri Harischandra, Thamira Weerakoon, and Randil Pushpananda</i>	

Computational Paralinguistics

Spoken Emotion Recognition Using Soft Labels	101
<i>Dániel Halmai and Gábor Gosztolya</i>	
NAMTalk: From Muscle Vibrations to Emotional Speech	113
<i>Kunjan Gajre, Rajnidhi Gupta, Ravindrakumar M. Purohit, and Hemant A. Patil</i>	
What Do LLMs Know About Human Emotions? The Russian Case Study	129
<i>Olga Mitrofanova, Polina Iurevtseva, and Maxim Bakaev</i>	

Emotions Manifestation by Adolescents with Intellectual Disabilities 145
Egor Kleshnev and Elena Lyakso

Retention-Augmented Voice Assistant: A Lightweight Architecture
for Stateful Interaction with Comprehensive Evaluation
and Privacy-Preserving Design 157
*Abdelkader Seif El Islem Rahmani, Yasser Yahiaoui,
and Abdelghani Bouziane*

Speech Processing for Healthcare

Investigation of Explainable Multimodal Methods for Detecting Mental
Disorders 173
Mikhail Dolgushin, Daria Guseva, and Alexey Karpov

Attention Deficit Hyperactivity Disorder: Identifying Approaches
for Early Diagnosis, a Pilot Study 188
*Elena Lyakso, Olga Frolova, Anton Matveev, Petr Shabanov,
Andrei Lebedev, Aleksandr Nikolaev, Egor Kleshnev,
Severin Grechanyi, and Ruban Nersisson*

Text-to-Dysarthric-Speech Generation for Dysarthric Automatic Speech
Recognition: Is Purely Synthetic Data Enough? 203
Wing-Zin Leung, Heidi Christensen, and Stefan Goetze

Colour Preferences in Schizophrenic Speech 217
Anna Shevlyakova, Vladimir Bochkarev, and Stanislav Khristoforov

Automated Assessment of Phrase Intelligibility for Russian Speech Based
on Esophageal Voice 228
Evgeny Kostyuchenko

Speech and Language Resources

Subtle Changes in L1 Stops of Late Salento Italian-French Bilinguals:
An Acoustic Study Using AutoVOT Adapted for Italian and French 241
Marie Fongaro, Barbara Gili Fivela, Maud Pélissier, and Gabriel Hévr

Sound and Colour in Phonosemantics: Perceptual and Acoustic Correlates
of Mongolian Vowels 256
*Rodmonga Potapova, Vsevolod Potapov, Tsend-Ayush Ganbaatar,
Leonid Motovskikh, and Nikolay Bobrov*

Rhythmic Diglossia Based on Discourse Types and Dialects of English: Australian and New Zealand Corpora	267
<i>Anna Borzykh and Tatiana Shevchenko</i>	
Automatic Annotation of Discourse and Speech Formulas in Internet Communication: A Telegram Comment Corpus	278
<i>Aleksandra S. Maslenikova and Tatiana I. Popova</i>	
Speaker Recognition	
Effect of Spoof Speech on Forensic Voice Comparison Using Deep Speaker Embeddings	295
<i>Mohammed Hamzah Alsalihi and Dávid Sztahó</i>	
Source Vendor Tracing of Audio Deepfakes	307
<i>Marina Volkova, Artem Chirkovskiy, Egor Ausev, and Ekaterina Shangina</i>	
Language-Specific Adaptation Strategies for Speaker Recognition Using MobileNet	322
<i>Anton Yakovenko, Evgeny Bessonitsyn, Valeria Efimova, and Mark Zaslavskiy</i>	
Enhancing Audio Replay Attack Detection with Silence-Based Blind Channel Impulse Response Estimation	333
<i>Şule Bekiryazıcı, Cemal Hanilçi, and Neyir Özcan</i>	
Author Index	345

Automatic Speech Recognition



In-Domain SSL Pre-training and Streaming ASR: Application to Air Traffic Control Communications

Jarod Duret¹, Salima Mdhaffar¹, Gaëlle Laperrière¹, Ryan Whetten¹,
Audrey Galametz², Catherine Kobus², Marion-Cécile Martin², Jo Oleiwan²,
and Yannick Estève¹(✉)

¹ Avignon Université - LIA, Avignon, France

yannick.esteve@univ-avignon.fr

² AIRBUS, Blagnac, France

Abstract. In this study, we investigate the benefits of domain-specific self-supervised pre-training for both offline and streaming ASR in Air Traffic Control (ATC) environments. We train BEST-RQ models on 4.5k hours of unlabeled ATC data, then fine-tune on a smaller supervised ATC set. To enable real-time processing, we propose using chunked attention and dynamic convolutions, ensuring low-latency inference. We compare these in-domain SSL models against state-of-the-art, general-purpose speech encoders such as w2v-BERT 2.0 and HuBERT. Results show that domain-adapted pre-training substantially improves performance on standard ATC benchmarks, significantly reducing word error rates when compared to models trained on broad speech corpora. Furthermore, the proposed streaming approach further improves word error rate under tighter latency constraints, making it particularly suitable for safety-critical aviation applications. These findings highlight that specializing SSL representations for ATC data is a practical path toward more accurate and efficient ASR systems in real-world operational settings.

Keywords: Speech recognition · Self-supervised learning · Streaming · Air traffic control

1 First Introduction

Automatic speech recognition (ASR) has become an essential technology in various fields, including aviation, where precise and real-time transcription of spoken communication could become crucial [9]. Air Traffic Control (ATC) communications represent a particularly challenging domain for ASR due to their constrained but highly specialized vocabulary, strict grammar structures, and wide range of speaker accents. These factors, combined with the presence of background noise, make ATC a specific and demanding application for speech recognition systems, especially when real-time processing is targeted. While SSL-pre-trained models such as wav2vec 2.0 [2] and HuBERT [11] have demonstrated

strong performance across various ASR benchmarks, their adaptation to highly specialized domains like ATC presents unique challenges that require further investigation [20]. A key consideration for existing SSL models is that they are often pre-trained on diverse but predominantly general-purpose speech data, which may not fully align with the linguistic and acoustic characteristics of specialized domains like ATC.

In this work, we explore the impact of in-domain SSL pre-training for offline and streaming ASR dedicated to ATC communications. To evaluate our approach, we conduct experiments on both proprietary and publicly available ATC datasets. This paper is organized as follows: first, we discuss related work and the characteristics of ATC communications and datasets. Next, we describe the pre-training process of our SSL models and compare their performance for offline ASR. We then introduce a method for pre-training a model for streaming ASR and present experimental results.

2 Related Work

The application of self-supervised learning (SSL) models to ATC speech processing has gained significant attention in recent years. In [20], the authors investigated the suitability of pre-trained SSL wav2vec 2.0 models, for transcribing ATC speech and detecting key information, demonstrating that these models can benefit from domain adaptation techniques to improve recognition performance in this specialized setting. In addition to improving accuracy, recent research has also focused on optimizing SSL models for real-time streaming applications [12], which is crucial for ATC scenarios where latency should be minimized for grounded scenarios. Another key development in SSL-based streaming ASR is the BEST-RQ model [5], which introduces quantization techniques for improved representation learning. Originally designed to enhance efficiently self-supervised learning with discrete latent representations, BEST-RQ has also been explored for its potential in streaming speech recognition [5].

3 ATC Datasets Used in This Study

The Airbus-ATC corpus is the dataset released for the Airbus ATC Speech Recognition 2018 Challenge [14]. A software-defined radio receiver connected to an aeronautical antenna and set to capture local airport ATC communications has been used to record the audio at 16 kHz [7]. Since the collected audios originate from French airports, the French accent is predominant. The Airbus-ATC corpus contains approximately 50 h of recorded communications gathered from multiple French airports, split into 3 datasets: about 40 h for supervised training, 5 h for validation, and 5 h for evaluation.

The ATCO2 corpus [8] contains also real-world ATC voice recordings. It brings together thousands of unlabelled hours of communications between air-traffic controllers and pilots, drawn from publicly available sources such as

LiveATC, and a small part (1 h for the free version, 4 h for the purchased version) of manually transcribed speech recording. These recordings vary widely in audio quality, airport environments, and speaker accents, capturing the realities of high-stakes aviation dialogue. While the Airbus-ATC corpus is mainly French accented, the ATCO2 corpus contains several accents, mainly Czech, Swiss German, and also Swiss French and Australian English.

In the context of ATC, messages can be categorized into three main types. First, communications from the air traffic controller serve as authoritative instructions or clearances directed to pilots. Second, pilot transmissions function as responses or requests for clarification, position reports, or emergency declarations, facilitating coordination with air traffic controllers and ensuring adherence to given instructions. These two kinds of messages, from control agents and pilots, are present in both the Airbus-ATC and ATCO2 corpora. Finally, the Automatic Terminal Information Service (ATIS) provides continuous updates on meteorological conditions, runway availability, and operational notices relevant to a specific airport. ATIS is characterised by utterances of about 30 s in average, longer than regular exchanges between pilot and the ATC, which have an average duration of 4.5 s. ATIS messages are present in Airbus-ATC, but not in the ATCO2 corpus.

In our study, we used around 4,500 h of ATCO2 unlabelled audio recordings in English for SSL pre-training, the official distribution of the Airbus-ATC 2018 challenge for ASR supervised fine-tuning (40 h) and evaluation, the freely available ATCO2-test-1h¹ and the licensed ATCO2-test-4h for evaluation only.

4 In-Domain and Out-Domain SSL Models

4.1 Self-supervised Learning of in-Domain Models

To pre-train in-domain models via self-supervision, we selected the BEST-RQ framework. This decision was influenced by its open-source availability within the Speechbrain project [16] and its efficiency—2.5 times faster than wav2vec 2.0 [18]. In addition, it demonstrates performance comparable to the widely used wav2vec 2.0 approach [5, 18]. BEST-RQ is a self-supervised learning approach that uses a random-projection quantizer to turn speech signals into discrete labels, then trains a speech encoder to predict those labels for masked parts of the input. Because the quantizer is fixed and untrained, it places fewer constraints on the encoder architecture—allowing both streaming and non-streaming models—and avoids the added complexity of jointly learning a representation [5].

For pre-training, we rely on the SpeechBrain recipe² applied to 4,500 h of unlabelled ATCO2 English audio. We trained a *Large* model of 300M parameters with 848 dimensions for hidden representations and 24 encoder layers. This model is pre-trained for 300K iterations by employing sixteen H100 GPUs. We

¹ https://huggingface.co/datasets/Jzuluaga/atco2_corpus_1h.

² <https://github.com/speechbrain/speechbrain/tree/develop/recipes/LibriSpeech/self-supervised-learning/BEST-RQ>.

select the batch size to optimize GPU memory usage, resulting in 2 h of audio per batch. Training this *Large* BEST-RQ model for 300K iterations requires approximately two days. The masking strategy uses segments of four frames with a probability of 0.15, meaning that 15% of segments are masked (i.e. 60% of speech).

This model is called BRQ-ATCO2_{Large}.

4.2 Out-Domain Existing SSL Models

In order to compare our in-domain SSL models to existing out-domain models on the ASR task applied to ATC data, we made some experiments (described in Sect. 5) by using several of the most popular ones: wav2vec2.0 models (XLSR-128 [1] and LS960 [2]), MMS-1B [15], wavLM [4], HuBERT [11], and w2v-BERT 2.0 [3, 6]. In this paper, we focus on the two speech encoders that delivered the best results on our preliminary experiments as the other models had a significantly higher word error rate (WER). The two models we kept for this paper are HuBERT [11] and w2v-BERT 2.0 [3, 6]. The HuBERT Large model used in this study has been pre-trained on 60,000 h of unlabelled English speech from the Libri-Light dataset while w2v-BERT 2.0 has been pre-trained on 4.5 million hours of speech in 143 languages from diverse public datasets.

Table 1. ASR results of the two best out-domain speech encoders compared to our in-domain speech encoder on the Airbus-ATC dev and test corpora, and on the two ATCO2 test corpora.

SSL model	Pre-train	#Par.	LM	A-dev	A-test	AT2-1h	AT2-4h
w2v-BERT 2.0	4.5M	600M	4-gram	6.74	6.21	23.12	29.30
HuBERT	60k	300M	4-gram	7.70	7.25	33.33	39.26
BRQ-ATCO2 _{Large}	4.5k	300M	4-gram	7.90	7.40	19.70	28.70
w2v-BERT 2.0	4.5M	600M	—	7.36	7.01	25.98	31.74
HuBERT	60k	300M	—	8.93	8.50	37.55	43.39
BRQ-ATCO2 _{Large}	4.5k	300M	—	10.67	10.01	24.27	30.73

5 Offline ASR on the ATC Data

To compare the in-domain BRQ-ATCO2_{Large} model described in Sect. 4.1 with popular out-domain SSL models, we fine-tune these models for ASR applied to ATC data. As mentioned in Sect. 3, we use the 40 h of the labelled Airbus-ATC training data for these fine-tunings, in addition to the 5 h for development purposes.

5.1 Offline ASR Fine-Tuning Setup

For the fine-tuning phase, we adopt as a downstream probe a straightforward architecture consisting of a 3-layers DNN followed by a linear layer and a softmax activation function. The training is performed using the Connectionist Temporal Classification (CTC) loss. The probe’s hidden layers have a dimension of 1024, with a dropout rate of 0.15.

We use different learning rates and optimizers for the pre-trained encoder and the probe. We fine-tune the BEST-RQ encoder using a learning rate of 10^{-4} while the probe is trained with a higher learning rate of 8×10^{-4} . w2v-BERT is fine-tuned using a learning rate of 1×10^{-5} with a probe learning rate of 1.5. HuBERT is fine-tuned using a learning rate of 1×10^{-4} with a probe learning rate of 1.0. The batch size is adjusted according to the encoder size to fit within an A100 80GB GPU, resulting in approximately 450 s of audio per batch for *Large* BEST-RQ and HuBERT models and 40 s for w2v-BERT. We fine-tune the entire model for 30 epochs with BEST-RQ and 80 epochs with HuBERT and w2v-BERT on Airbus-ATC training set, selecting the best checkpoint based on the WER obtained on Airbus-ATC development set. A 4-gram language model (LM) was trained on the Airbus-ATC training data, ATCOSIM [10] and UWB_ATCC [17] datasets. We report on results with and without this 4-gram LM. When the LM is used we use beam search decoding with a beam size of 1000. When no LM is applied, the model defaults to greedy decoding.

5.2 Experimental Results

Table 1 presents the word error rate (WER) obtained on the Airbus-ATC and ATCO2 test sets by using the two existing out-domain SSL models we selected in regards with their performance and our in-domain SSL model, with and without language model integration. w2v-BERT 2.0, with 600M parameters and significant pre-training on 4.5 millions hours of multilingual speech, demonstrates superior performance on Airbus datasets. It achieves a WER of 6.21% on the Airbus-ATC test set and 23.12% on the ATCO2-1h subset when using a 4-gram language model. The HuBERT model, with its smaller architecture of 300M parameters pre-trained on 60k hours of English speech, shows competitive but slightly inferior performance, achieving 7.25% and 33.33% WER on the same subsets respectively. Interestingly, BRQ-ATCO2_{Large}, despite its architecture of 300M parameters and limited pre-training on 4.5k hours of ATCO2 unlabelled data, achieves the best performance on the ATCO2-1h and ATCO2-4h subsets with 19.7% and 8.7% WER, significantly outperforming both w2v-BERT 2.0 and HuBERT. While both Airbus-ATC and ATCO2 datasets contain air traffic control communications, they present distinct acoustic characteristics. The ATCO2 corpus was collected through a network of very-high frequency (VHF) radio receivers operated by volunteers, resulting in specific acoustic conditions influenced by factors such as equipment quality (various antenna types and SDR receivers), signal reception, and environmental variables. Additionally, as mentioned in Sect. 3, Airbus-ATC and ATCO2 do not contain the same accents.

BRQ-ATCO2_{Large}, being pre-trained specifically on this type of data, demonstrates particularly strong performance on the ATCO2 corpus, highlighting the importance of domain-specific pre-training for handling specialized acoustic conditions.

We observe that the integration of a language model consistently improves performances across all models. This improvement is even more pronounced for BRQ-ATCO2_{Large}, particularly on the ATCO2-1h corpus, where the WER decreases by 4.57 points (from 24.27% to 19.70%). These results suggest that while extensive pre-training can be beneficial for robustness, as demonstrated by w2v-BERT 2.0 performance on Airbus data, our in-domain pre-training approach with BEST-RQ is effective for specific ATC contexts, despite using significantly fewer hours.

6 Streaming ASR for ATC

6.1 Streaming Self-supervised Learning

To implement a streamable version of BEST-RQ, we replace classical attention mechanisms in the Conformer blocks by chunked attention [19]. This approach divides the input sequence into chunks that group a given amount of frames. Within each chunk, frames can attend to all other frames in the same chunk. Additionally, chunks can attend to a limited number of previous chunks. We also integrate Dynamic Chunk Convolutions (DCConv) [13] instead of conventional convolution layers in the Conformer blocks and reuse the same chunk boundaries we used for chunked attention. Unlike conventional convolutions, which create a mismatch between training and inference due to access to future context beyond chunk boundaries, DCConv restricts the convolution operation to within-chunk frames.

With the purpose of developing a model that can flexibly adapt to different streaming requirements at inference time, we implement a mixed training strategy: 40% of batches are trained with full context, without any chunking constraints, and 60% of batches use dynamic chunking. Chunk size is randomly sampled between 8 and 32 frames, and 75% of chunks have restricted left context (2–32 chunks), while 25% maintain full left context. This mixed training approach results in a model that can operate across different latency requirements at inference time, from low-latency streaming to full-context processing.

We train two BEST-RQ models with our streaming approach: a *Large* version with the same size as the one trained in classical offline mode, called Stream-ATCO2_{Large}, and a *Base* one with 92M parameters, called Stream-ATCO2_{Base}. To move from *Large* to *Base*, we decrease the dimension of hidden representations from 848 to 576 and reduce the number of encoder layers from 24 to 12. Both models are pre-trained with 300k iterations. The *Base* model is trained on four H100 GPUs with 1.6 h of audio per batch while the *Large* keeps the same settings as for its offline version.

6.2 Streaming ASR Fine-Tuning Setup

For the fine-tuning phase in streaming mode, we reuse the probe architecture described in Sect. 5. We modify the supervised learning strategy by applying dynamic chunking to 100% of the batches, rather than the mixed approach used during the SSL pre-training.

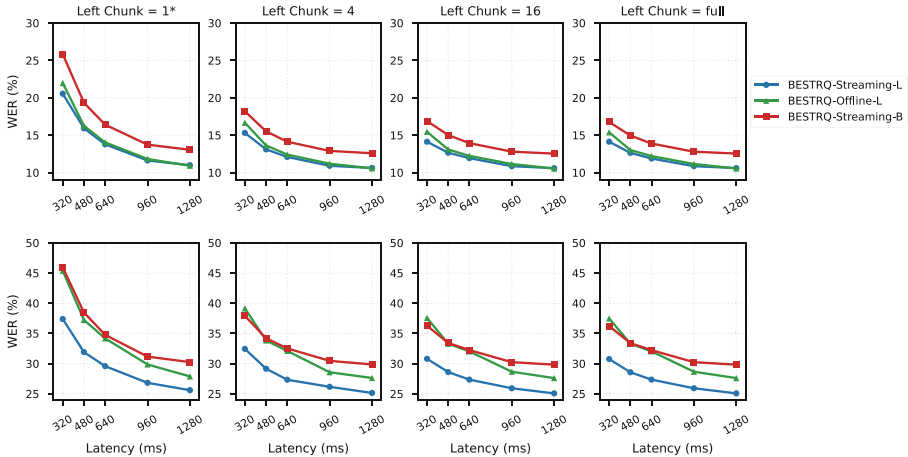


Fig. 1. Performances of BEST-RQ encoders fine-tuned on the **streaming** ASR task. (top) Results on the Airbus-ATC test data. (bottom) Results on the ATCO2-1h test data. Different left contexts and (right) chunk sizes are investigated.

6.3 Experimental Results on Streaming ASR

The streaming capabilities of our BEST-RQ streaming models on both Airbus-ATC and ATCO2 test sets are shown in Fig. 1. We compare three variants of SSL models: Stream-ATCO2_{Large}, BRQ-ATCO2_{Large} (pre-trained without streaming strategy), and Stream-ATCO2_{Base} models. Figure 1 presents the WER results across different left context sizes and chunk sizes, which correspond to controlled model’s latency. No language model was integrated. On the Airbus-ATC test set, Stream-ATCO2_{Large} achieves the best performance across all configurations, ranging from 10.6% to 20.55% depending on streaming settings. With full left context, it reaches a WER of 10.6% at a 1280ms latency, showing minimal degradation compared to the offline model (10.6%). With aggressive streaming constraints (left chunk = 1), which has not been seen during training, the model achieves 20.55% WER at 320ms of latency, demonstrating the robustness of the model in low-latency scenarios. The impact of streaming adaptation is evident on the ATCO2 dataset, where the streaming fine-tuning shows substantial improvements. The Stream-ATCO2_{Large} model outperforms its offline counterpart by a fair margin, achieving 25.07% WER versus 27.6% WER at 1280ms

of latency with full left context. The performance gap between the two models is even bigger at lower latencies. For both datasets, larger left context sizes improve performance, but improvements become minimal beyond 16 chunks. Finally, the Stream-ATCO2_{Base} model shows competitive performance, particularly on the ATCO2 dataset when compared to the offline BRQ-ATCO2_{Large} model fine-tuned for streaming ASR. These results highlight the viability of the *Base* architecture for applications where computational resources are limited, and show that our streaming SSL and fine-tuning strategies effectively adapt the BEST-RQ models for real-time ASR applications for ATC data, particularly for strict latency constraints.

6.4 SSL Streaming Models Applied to Offline ASR

We also tested the performance of our models pre-trained through our streaming SSL approach but used for ASR in offline mode, without latency constraints. In this case, we fine-tuned the models following the offline mode described in Sect. 5.1.

Despite restricting the context in pre-training, we can observe that the *Large* model pre-trained in streaming mode outperforms in WER the offline BRQ-ATCO2_{Large} model pre-trained in a conventional way in all the test datasets (Table 2), even outperforming the HuBERT model (see Table 1).

Table 2. WER of the different BEST-RQ models and the Airbus-ATC and ATCO2 test datasets for **offline** ASR with the a 4-gram language model. Pre-training in a streaming fashion by restricting context, proved to be helpful even in an **offline** (non-streaming) ASR setting.

SSL model	Airbus	ATC02-1h	ATC02-4h
BRQ-ATCO2 _{Large}	7.40	19.70	28.70
Stream-ATCO2 _{Large}	7.18	19.30	26.59
Stream-ATCO2 _{Base}	8.09	24.58	29.47

To further analyze the performance of the streaming SSL model on the offline ASR task, we computed the WER based on the type of messages (see Sect. 3). Table 3 presents the WER for controller, pilot, and ATIS messages in the Airbus-ATC test corpus. An improvement is observed for each type of message, with the smallest occurring in ATIS messages (−1.37% relative) and the largest in the noisiest category, pilot messages (−3.46% relative). These results suggest that the mixed SSL training approach using both full context samples and dynamic chunking is particularly useful to process noisy recordings.

Table 3. WER according to speaker role (C:controller, P:pilot, A:ATIS) on the Airbus-ATC test corpus for **offline** ASR with a 4-gram language model.

SSL model	C	P	A
BRQ-ATCO2 _{Large}	4.84	10.41	5.84
Stream-ATCO2 _{Large}	4.73	10.05	5.76

7 Discussion

In this work, we investigated how in-domain SSL pre-training impacts both offline and streaming ASR for Air Traffic Control (ATC) communications. We found that while large-scale, general-purpose models (e.g., w2v-BERT 2.0 with 4.5M hours) excel at broad tasks, our BEST-RQ model—which uses only 4.5k hours of domain-specific data—outperforms them on the ATCO2 corpus. This shows that targeted in-domain pre-training can be more effective than large-scale general-purpose training for the unique acoustic conditions of VHF radio communications. We also demonstrated that our streaming strategy—combining chunked attention, dynamic convolutions, and a mixed training approach using both full context samples and dynamic chunking—is highly effective for real-time ATC speech recognition. Notably, this approach also proves beneficial for offline ASR. Our BEST-RQ Large model pre-trained in streaming mode retains strong performance across different latency settings, with only minimal degradation compared to its offline counterpart. Even under strict latency limits, the model remains practical. This model also achieves better offline ASR performance than its counterpart pretrained using a conventional offline approach. Overall, these results highlight the value of specialized SSL pre-training combined with a streaming approach for focused speech recognition tasks. Future work could involve integrating these models into operational ATC systems and examining their robustness to the diverse accents and noise conditions common in ATC communications.

Acknowledgments. This work used HPC resources from GENCI-IDRIS: grants AD0-11012551R3, AD011015051R1, AD011012108R3, AD011014814R1, and AD011015509.

References

1. Babu, A., et al.: Xls-r: Self-supervised cross-lingual speech representation learning at scale. arXiv preprint [arXiv:2111.09296](https://arxiv.org/abs/2111.09296) (2021)
2. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. **33**, 12449–12460 (2020)
3. Barrault, L., et al.: Seamless: multilingual expressive and streaming speech translation. arXiv preprint [arXiv:2312.05187](https://arxiv.org/abs/2312.05187) (2023)
4. Chen, S., et al.: Wavlm: large-scale self-supervised pre-training for full stack speech processing. IEEE J. Sel. Topics Signal Process. **16**(6), 1505–1518 (2022)

5. Chiu, C.C., Qin, J., Zhang, Y., Yu, J., Wu, Y.: Self-supervised learning with random-projection quantizer for speech recognition. In: International Conference on Machine Learning, pp. 3915–3924. PMLR (2022)
6. Chung, Y.A., et al.: W2v-BERT: combining contrastive learning and masked language modeling for self-supervised speech pre-training. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 244–250. IEEE (2021)
7. Delpuch, E., et al.: A real-life, French-accented corpus of air traffic control communications. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018). European Language Resources Association (ELRA), Miyazaki, Japan (2018). <https://aclanthology.org/L18-1453/>
8. Gomez, J.P.Z., et al.: ATCO2 corpus: a large-scale dataset for research on automatic speech recognition and natural language understanding of Air Traffic Control communications. J. Data-centric Mach. Learn. Res. (2024)
9. Helmke, H., Ohneiser, O., Mühlhausen, T., Wies, M.: Reducing controller workload with automatic speech recognition. In: 2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC), pp. 1–10. IEEE (2016)
10. Hofbauer, K., Petrik, S., Hering, H.: The atcosim corpus of non-prompted clean air traffic control speech. In: LREC. Citeseer (2008)
11. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhota, K., Salakhutdinov, R., Mohamed, A.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. IEEE/ACM Trans. Audio Speech Lang. Process. **29**, 3451–3460 (2021)
12. Kanagawa, H., Ijima, Y.: Knowledge distillation from self-supervised representation learning model with discrete speech units for any-to-any streaming voice conversion. In: Proceedings of Interspeech 2024, pp. 4393–4397 (2024)
13. Li, X., Huybrechts, G., Ronanki, S., Farris, J., Bodapati, S.: Dynamic chunk convolution for unified streaming and non-streaming conformer asr. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
14. Pellegrini, T., Farinas, J., Delpuch, E., Lancelot, F.: The Airbus Air Traffic Control speech recognition 2018 challenge: towards ATC automatic transcription and call sign detection. In: 20th Annual Conference of the International Speech Communication Association (INTERSPEECH 2019), pp. 2993–2997 (2019)
15. Pratap, V., et al.: Scaling speech technology to 1,000+ languages. arXiv (2023)
16. Ravanelli, M., et al.: Open-source conversational AI with Speechbrain 1.0. J. Mach. Learn. Res. **25**(333), 1–11 (2024)
17. Šmídl, L., Švec, J., Tihelka, D., Matoušek, J., Romportl, J., Ircing, P.: Air traffic control communication (atcc) speech corpora and their use for asr and tts development. Lang. Resour. Eval. **53**, 449–464 (2019)
18. Whetten, R., Kennington, C.: Open implementation and study of BEST-RQ for speech processing. In: IEEE ICASSP 2024 Workshop on Self-Supervision in Audio, Speech and Beyond (SASB 2024) (2024)
19. Zhang, B., et al.: Unified streaming and non-streaming two-pass end-to-end model for speech recognition. arXiv preprint [arXiv:2012.05481](https://arxiv.org/abs/2012.05481) (2020)
20. Zuluaga-Gomez, J., et al.: How does pre-trained wav2vec 2.0 perform on domain-shifted ASR? An extensive benchmark on Air Traffic Control communications. In: 2022 IEEE Spoken Language Technology Workshop (SLT), pp. 205–212. IEEE (2023)



Evaluating the Performance of Several ASR Systems in Environmental and Industrial Noise

Sara M. Pearsell¹  , Oliver Niebuhr¹ , and Samuel Schmück² 

¹ University of Southern Denmark, 6400 Sønderborg, Denmark
{pearsell,olni}@sdu.dk

² Lancaster University, Bailrigg, Lancaster LA1 4YL, UK
s.schmueck@lancaster.ac.uk

Abstract. Automatic Speech Recognition (ASR) systems are becoming more commonplace in real-world applications. Despite this increase in usage, their robustness in noisy environments remains problematic for correct word identification. This study offers an automated program to test ASR systems alongside different background noise. It tests several ASR systems (Whisper-Small, Whisper-Medium, Whisper-Large-V3-Turbo, Parakeet 0.6b, Canary 1b, and Commonvoice-Wav2Vec-EN) across five total noise conditions (white noise, speech shaped noise, and three industrial noises) at varying levels of loudness (64–79 dB). Results indicate that ASR systems have significantly reduced word recognition across all noise levels with industrial machine noise posing a greater challenge than other types of noise at moderate intensities. Additionally, opting to avoid enhancements to ASR improved performance overall, particularly for female speech.

Keywords: Automatic Speech Recognition · Industrial Noise · Acoustic Signal Processing

1 Introduction

As technology continues to become increasingly integrated into everyday life so does its role in facilitating efficient and intuitive human-machine interaction (HMI). One advantageous area for development is voice-based interactions, where Automatic Speech Recognition (ASR) systems are positioned as relevant and important tools for facilitating seamless communication between users and machines [1]. ASR technologies have the potential to improve accessibility, streamline processes, and enhance automation across a wide range of applications including those found within industrial environments [2–5].

In manufacturing and other industrial settings, voice-controlled interfaces could drastically improve the operation efficiency of a company. By allowing workers to rely on spoken inputs rather than manual ones (i.e. typing commands or pressing buttons), employees could interact with machine while remaining mobile. This would mean no longer having the constraints of being confined to fixed stations or being required to engage in physical maneuvering of controls such as pressing a button to initiate or



Fig. 1. Image generated by OpenAI’s ChatGPT with DALL E, April 2025.

stop a machine. Figure 1 illustrates to current physical constraints of manually operate machines in factory settings.

Despite their promise, the applications of ASR system in industrial environments do face two major challenges. For one, there is a notable absence of real-world implementation or of the systematic evaluation of ASR technologies in more acoustically complex, temporally variable, and high-noise level, industrial environments like factory halls. There is also the problem that existing ASR models are typically trained on clean or lightly noisy speech data, therefore lacking the robustness for the unique properties of industrial noises like those of machinery. Notably, there also appears to be no publicly available audio corpus representing authentic industrial sounds for training or benchmarking ASR performance.

The present study is a part of the larger Arrowhead Tools initiative, a European project aimed as advancing automated, digitized engineering tools for industry [6]¹. The EU-funded Arrowhead project develops a service-oriented platform to enable interoperability and plug-and-play integration in industrial automation. It aims to standardize interfaces for improved efficiency, flexibility, and cost-effectiveness in smart manufacturing, energy systems, mobility, and other domains. One of our key contributions in this initiative the development of an acoustically robust command inventory which can reliably function in the presence of noisy industry environments. This includes identification of speech

¹ <https://fpvn.arrowhead.eu/fpvn-arrowhead/>

features which are most resistant to acoustic masking and understanding the differences in ASR structures' responses to adverse noise conditions.

Previous research has demonstrated that ASR performance degrades significantly when additional noise is present in the acoustic signal. Numerous studies have documented consistent reductions in recognition accuracy due to background sounds such as street traffic, white noise, and speech babble [7]. Speech-shaped noise, which closely mimics the spectral characteristics of human speech, has been shown to substantially impair word recognition [8]. Many other studies have systematically examined how types of background noise affect word recognition accuracy [9–13].

However, while this body of work provides a valuable foundation, it can create the misleading impression that the challenges of ASR in noisy environments have already been exhaustively studied and solved. This is far from the case. Existing research has largely focused on noise types typical of public or conversational settings, leaving one critically important category almost completely unexamined: industrial machine noise. Our own previous work [14, 15] has demonstrated that machine noise introduces entirely different acoustic challenges for ASR systems, resulting in distinct patterns of recognition errors and degraded performance that cannot be predicted from results with other noise types. Therefore, assuming that an ASR system, which performs effectively under street or speech-babble noise, will generalize equally well to industrial noise is not only unjustified but potentially counterproductive for deployment in automation contexts. Addressing this gap—by systematically investigating ASR performance under complex, variable, and largely unexplored machine-noise conditions—is the central motivation of our line of research and the present study. Although we acknowledge ASR architectures, including the larger models like Whisper-Large-V3 and Parakeet, are computationally expensive and other more lightweight/task specific models may be more appropriate for our current experimental paradigm, we are interested in addressing the gap. Our present study offers the opportunity to investigate the potential benefits or pitfalls of ASR systems within industry noise including important aspects like acoustic thresholds.

In our prior research [14, 15] we evaluated two widely available commercial ASR systems, Apple Dictation and Google Translate, under several types of environmental noise. The present study builds on that work by accessing multiple open-source ASR models (Whisper-Small, Whisper-Medium, Whisper-Large-V3-Turbo, Parakeet 0.6b, Canary 1b, and Commonvoice-Wav2Vec-EN) under five different types of background noise including white noise, speech shaped noise, and three varieties of industrial machine noise. Each condition was tested across a range of sound pressure levels (67 dB to 79 dB) to simulate realistic levels found on factory floor environments.

We used a previously developed list of nonwords as the basis of our novel command inventory designed to be robust in acoustic noise using various linguistic premises and information from some of the previous literature on ASR and noise. Nonwords are words which could exist, as in they follow the sound and syllable structures of a given language, but do not actually exist in language. However, ASR systems require valid lexical inputs, i.e., real words. We selected 17 nonwords from our command inventory list which is a part of the broader research for the Arrowhead Project and selected real English words which were phonologically and phonetically like these nonwords. These words share consonant-vowel (CV) structure as well as similar acoustic-phonetic

properties. For example, the nonword *sisawp* would match with the English real word *sea salt* or *seesaw*. The current data thus not only illuminates the noise susceptibility of ASR systems but provides early insight into the which phonemics structures are most robust in adverse acoustic conditions. This information is imperative for improving and designing effective and noise resistant voice command inventories.

Our overarching objective is to understand how speech production and perception are affected by environments with excess noise and how to apply that knowledge to the design of efficient, user-friendly HMI systems for industrial settings. As automation continues to increase across European manufacturing sectors, particularly in high-cost domains like automotive production, the need for integration between human operators and automated systems becomes increasingly more important. Usage of effective voice-controlled interfaces can assist in lower production costs while also maintaining worker wellbeing, safety, and operational control.

However, achieving these goals requires overcoming some fundamental challenges which still exist in the design of speech-based interfaces. ASR systems need to function more reliably in acoustically diverse and multilingual environments, while support workers for both cognitive and environmental stresses. Our ongoing research in this field, including the current and future studies, is aimed at creating adaptive interaction strategies which reduced ASR error rates, minimize user frustration, and generally improve the overall workflow efficiency in noisy contexts.

In summary, this study is investigating how a variation of ASR system performs in acoustically adverse industrial environments with a focus on the intelligibility of stimuli under varying noise types and acoustic intensity. Through this, we are aiming to lay the groundwork for our future noise robust, phonetically informed command system designed to be intuitive, efficient, and accessible to many language backgrounds for HMI in noisy industrial environments like factory floors.

2 Data and Methodology

This section is divided into a description of our data collection and simulation methods, followed by a breakdown and discussion of our experimental setup establishing the methods used for the generation of our results.

2.1 Data Collection and Simulation

This study is interested in the performance and accuracy of ASR systems in environments with noise at varying intensities with a focus on industrial noise. While data collection is ongoing with the collection of speech samples through microphone towers in real industrial environments, we are interested in the potential to simulate this data for the purposes of preliminary study, and to determine the viability of these methods in the future compared to real industrial environments.

Two speakers (1 male, 1 female) were recorded in a sound treated booth using a ZOOM H6 with SSH 6 microphone producing 17 isolated speech segments (the target words). We aimed to investigate if any effects on word recognition exist due to gender differences. Previous research on gender within ASR systems found the average error

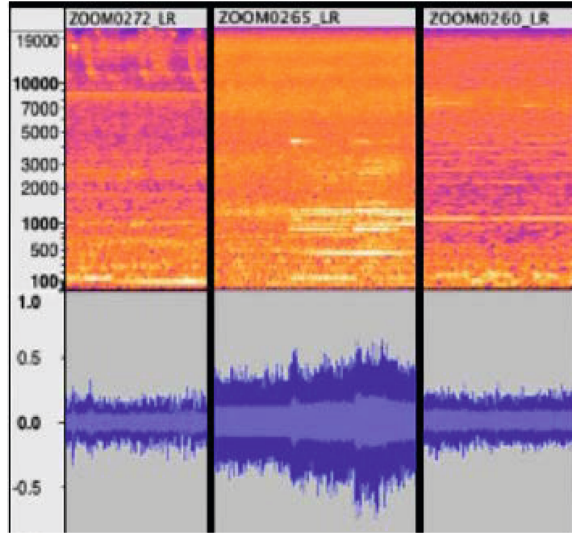


Fig. 2. Waveforms and spectrograms of the three industrial noises used in the ASR experiment. From left to right, the noises were a grinding machine (Planomat HP Blohm; ZOOM 0272), a cutting and polishing machine (MAM72-70V; ZOOM0265), and a milling and grinding machine (Röders TEC RXP500 076DSH; ZOOM0260).

rates for word recognition is significantly lower for female speakers than males [9, 10]. We are aiming to find if this result persists amongst other settings, like those in industrial environments. Since a large percentage of industrial environments employ male workers, understanding this potential implication of decreased work recognition accuracy with the incorporation of ASR systems could create problematic and costly outcomes.

The 17 speech segments 12 items within this inventory are singular tokens, 4 are compound nouns, and one item is a noun phrase with an indefinite determiner (e.g. a bowl). Each speaker produced each token resulting in a total of 34 files. Industrial noise recordings were collected in industrial environments and collected separately from speech recordings. Thus, resulting in five types of noise, speech noise, white noise, and three types of mechanical noise, the latter of which are shown in Fig. 2. We can clearly see in the figure that the noise of the cutting machine (middle) is not only the most intense, but also spectrally the most irregular and variable. In the frequency range up to 1 kHz, there is a high energy concentration of energy. Additionally, above this 1 kHz, the spectral slope (i.e. the successive loss of acoustic energy) is relatively shallow. For the grinder (left), too, most of the energy is concentrated in the frequency range up to 1 kHz. Unlike the cutting machine, the spectral energy drops relatively steeply after that, reaching a relatively low energy level already at about 3 kHz. The overall energy level of the milling machine (right) is like that of the grinder, but spectrally differently distributed. There is a clear spectral “gap” in energy between 500 Hz and 3 kHz, i.e., in a frequency range especially sensitive for speech production and perception.

We developed a Python program to automatically collect simulated data using our real-world recording set up which is available alongside the publication of this paper.

We used Wireshark with USBPcap to inspect the USB packets of our dB meter to replicate the USB exchange protocols within Python to allow the meter readings to be used within Python natively agnostic of operating system. Utilizing sounddevice and ffmpeg we built a lightweight program to calibrate each of our noise files to reach target mean A-weighted² dB (i.e. dBA [16]) values (67, 70, 73, 76, 79). Once calibrated the necessary coefficients to target each dBA level were preserved and used to control the system volume of each signal output.

Once the calibration results were calculated, we used speaker array consisting of two speakers to produce the noise samples at each target dBA from both left and right sides and a central, singular speaker array to produce speech at our target dBA of 70. Note that calibration parameters can be adjusted for future experimenters to replicate our results as well as conduct their own simulations using these methods. Producing these combinations of noise and speech samples resulted in 850 total simulated samples plus 34 samples produced without noise as a control. Each speech signal was produced to reach a mean dBA of 70. Speech samples were configured to play 1 s after initiating the noise signal and the noise sample was set to conclude after 3 s. Our final corpus is therefore technically time aligned and unifiable within our result to our initial phone level Praat [17] annotations, though it should be noted time alignment is seldom perfect in the context of automated playback and recording technologies without leveraging additional technologies. There are limitations to this approach in terms of the way this data is simulated which are appropriately explored within Sect. 5.1.

2.2 Experimental Setup

For our experiments 6 automatic speech recognition (ASR) and 1 automatic phone recognition (APR) systems were used for evaluation. An APR system was chosen to explore the accuracy of phone level transcription in noisy conditions, though it should be noted that APR technologies are often less accurate than their ASR counterparts due in part due to the increased resolution involved in phone level token classification. For ASR we used Whisper-Small, Whisper-Medium, Whisper-Large-V3-Turbo, Parakeet 0.6b, Canary 1b, and Commonvoice-Wav2Vec-EN. The quantised version of Whisper-Large was chosen as the accuracy was not found to be significantly different between the two models within a subsample of our data with the model consuming significantly less compute for inference.

In addition, two additional configurations of each of our 850 simulated noisy samples were generated using the metricgan-plus-voicebank and mtl-mimic-voicebank signal enhancement models to explore the viability of improving the clarity of noisy signals for ASR/APR input. For all compute an M2 Max Apple Silicon processor was used, whilst CUDA devices are often appropriately preferred for utilising such models on large scale datasets, for this scale of data and nature of the 3 s noisy simulated samples, we found this process to be appropriately powerful and advantageous for its use of

² For the assessment of medium-loud machine noises (60–80 dB), we chose the A-weighting as it robustly approximates the average human ear’s sensitivity at these moderate sound levels and, equally importantly, this choice is consistent with established standards for environmental and occupational noise assessments.

unified memory. Our results that we will now begin to discuss, involve the evaluation of the accuracy of our 7 speech technology models spanning our entire simulated corpus resulting in 18088 individual transcript results.

3 Results

A repeated-measures multivariate analysis of variance (RM-MANOVA) was conducted to investigate the effects of four independent variables on ASR target-word identification rates: Type of ASR (6 levels), Noise Type (5 levels), ASR Enhancement (3 levels), and Speaker Gender (2 levels). The dependent variable was the number of correctly identified target words, analyzed across five noise levels (67 dB to 79 dB in 3 dB increments).

3.1 General Performance

Overall, ASR target-word identification rates were relatively low across all noise levels, even at the lowest level of 67 dB³. Compared to commercial systems from Google and Apple previously evaluated [14, 15], the open-access systems tested here performed more poorly in general.

3.2 Effect of ASR Type

There were weak but generally consistent differences in target-word identification performance across the six ASR types. A statistically significant main effect of ASR Type emerged only at the 67 dB noise level ($F[5,2880] = 2.77$, $p = .018$), with effects at 70 dB and 79 dB approaching significance ($F[5,2880] = 2.14$, $p = .058$; $F[5,2880] = 2.10$, $p = .062$). The pattern underlying the main effect was consistent across levels: the ASR types *wav2vec2* and the whisper variants *medium* and *small* performed worst, while canary, parakeet, and whisper variant *large* yielded better results. Across all noise levels, parakeet achieved the highest average identification rate.

No consistent three-way or four-way interactions involving ASR Type were found. The only notable interaction was between ASR Type and Enhancement, where *not applying* enhancement (the NA condition) significantly improved target-word identification rates at 67 dB ($F[10,2880] = 2.12$, $p = .020$) and at 70 dB ($F[5,2880] = 2.03$, $p = .027$)—but only for the better-performing ASR types (canary, parakeet, and Whisper *large*). For *wav2vec2* and whisper variant *small*, there was no significant benefit from disabling enhancement, see Fig. 3.

³ Clean speech (i.e., 0 dB background noise) was not included, as the present study focuses exclusively on ASR behaviour in realistic noise conditions, where clean baselines are not relevant to the intended application context.

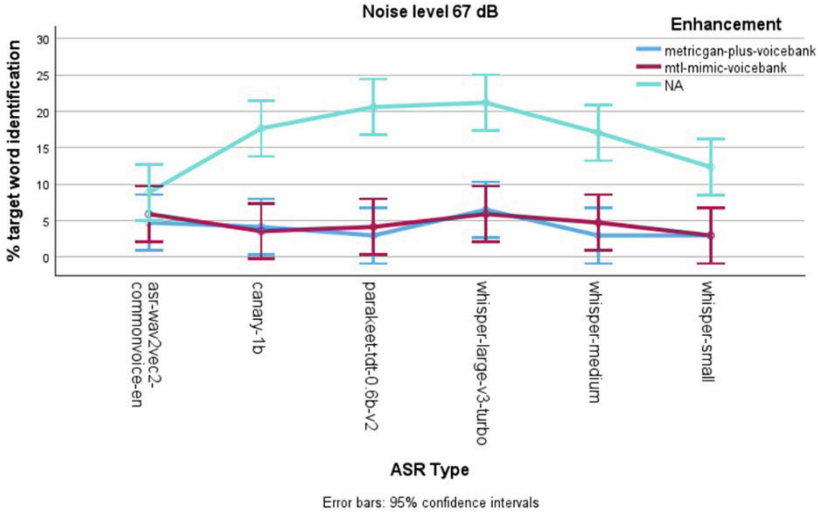


Fig. 3. Illustration of the ASR Type \times Enhancement interaction effects on target-word identification rate in %. $N = 510$ per data point.

3.3 Effect of Noise Type

Noise Type produced the strongest main effects across all five noise levels, explaining the most variance in the model. Up to 76 dB, target-word identification rates under speech babble noise were significantly higher—often by factors of two to four—than under any machine-noise type, highlighting the particular challenge industrial machine noises pose for ASR (67 dB: $F[4,2880] = 48.05$, $p < .001$; 70 dB: $F[4,2880] = 38.45$, $p < .001$; 73 dB: $F[4,2880] = 12.64$, $p < .001$; 76 dB: $F[4,2880] = 6.13$, $p < .001$; 79 dB: $F[4,2880] = 6.19$, $p < .001$). Even at 67 dB, identification rates under speech babble substantially exceeded those under all industrial noise types. Only the milling noise (ZOOM260) allowed for a few (but significantly, $p < 0.05$) more target words to be identified by the ASR systems than for the other machine-noise types, see. Figure 4 (top panel).

Interestingly, at the highest noise level (79 dB), this pattern reversed. All ASR systems performed worst under speech babble, while identification rates under ZOOM0260 (milling), ZOOM0265 (cutting), and ZOOM0272 (grinding) machine noises, though low, remained stable and significantly higher than for speech babble, see Fig. 4 (bottom panel).

This noise-level-dependent effect was further modulated by Enhancement. Disabling enhancement consistently increased identification rates under speech babble for levels below 79 dB, while at 79 dB (and likely above), disabling enhancement improved identification rates for all noise types except speech babble. All Noise Type \times Enhancement interactions were significant (67 dB: $F[8,2880] = 8.79$, $p < .001$; 70 dB: $F[8,2880] = 36.38$, $p < .001$; 73 dB: $F[8,2880] = 11.28$, $p < .001$; 76 dB: $F[8,2880] = 6.13$, $p < .001$; 79 dB: $F[8,2880] = 5.14$, $p < .001$).

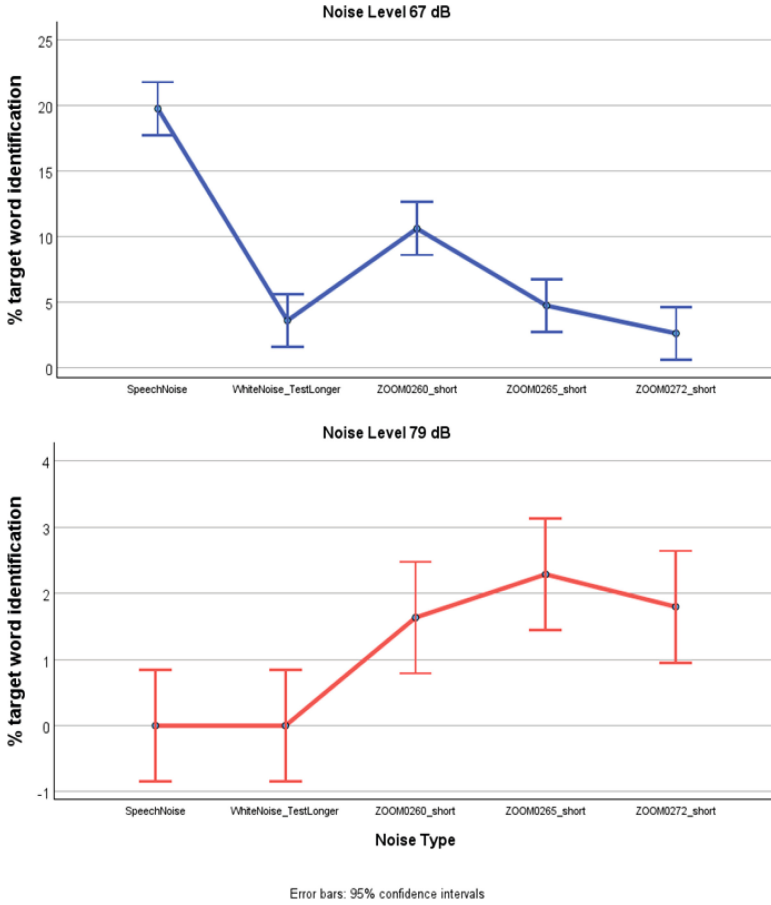


Fig. 4. Illustration of the main effect of Noise Type on target-word identification rate (in %) at the two extreme noise levels 67 dB (blue, top panel) and 79 dB (red, bottom panel). $N = 510$ per data point. (Color figure online)

3.4 Effect of ASR Enhancement

Consistent with these patterns, ASR Enhancement produced significant main effects at all noise levels except 79 dB. Across conditions, disabling enhancement improved target-word identification rates in all cases (67 dB: $F[2,2880] = 75.55$, $p < .001$; 70 dB: $F[2,2880] = 52.23$, $p < .001$; 73 dB: $F[2,2880] = 10.52$, $p < .001$; 76 dB: $F[2,2880] = 6.13$, $p < .001$).

The two enhancement types tested—metricgan-plus and mtl-mimic—performed similarly, with no statistically significant differences between them. However, both consistently reduced identification rates compared to the no-enhancement condition.

Notably, the benefit of disabling enhancement was especially strong for female-spoken target words. For males target words, enhancements status made only minor, often non-significant differences. In contrast, for female speech, enabling enhancement resulted in a substantial and consistent reduction in identification rates. Accordingly,

Enhancement \times Gender interactions were significant at almost all noise levels (67 dB: $F[2,2880] = 17.24$, $p < .001$; 70 dB: $F[2,2880] = 10.06$, $p < .001$; 76 dB: $F[2,2880] = 3.73$, $p < .001$).

3.5 Effect of Speaker Gender

Speaker Gender showed significant main effects for three of the five noise levels, with strong trends ($p < 0.1$) in the same direction for the other two. Overall, female target words were identified significantly more accurately (67 dB: $F[1,2880] = 76.07$, $p < .001$; 70 dB: $F[1,2880] = 7.27$, $p = .007$; 79 dB: $F[1,2880] = 15.22$, $p < .001$).

Beyond the Enhancement \times Gender interaction described above, Gender also showed consistent interactions with Noise Type (67 dB: $F[4,2880] = 3.88$, $p = .004$; 70 dB: $F[4,2880] = 2.55$, $p = .037$; 73 dB: $F[4,2880] = 7.09$, $p < .001$; 76 dB: $F[4,2880] = 2.72$, $p = .028$; 79 dB: $F[4,2880] = 7.74$, $p < .001$). These indicated that higher identification rates for speech babble as well as for the milling noise ZOOM260 (see Fig. 4) were primarily driven by the female speaker’s recordings, see Fig. 5. Similarly, improved recognition under industrial machine noise at 79 dB (see Fig. 4) was more pronounced for the female voice. This suggests that female speech may be generally more robust against masking by industrial background noise, or that its acoustic features facilitate better separation from noise for ASR systems.

No other consistent or significant two-way, three-way, or four-way interactions were observed.

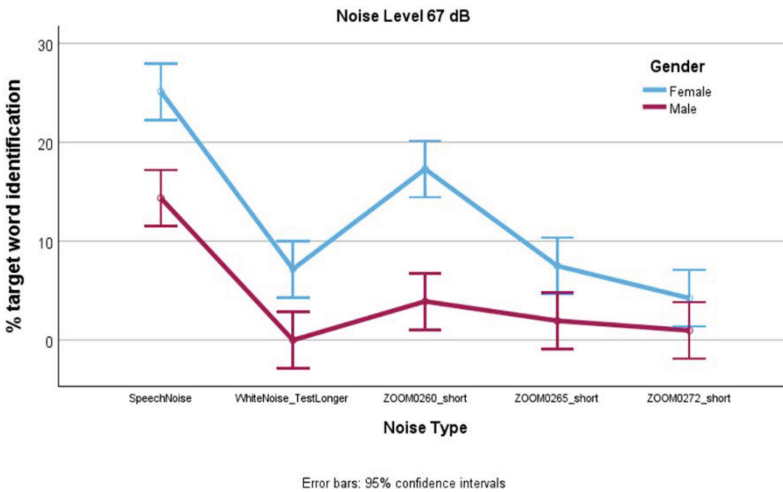


Fig. 5. Illustration of the Noise Type \times Gender interaction effects on target-word identification rate in %. $N = 510$ per data point.

3.6 Word-Specific Effects

Due to our focus on our core questions, we have not addressed word-specific effects in detail here. Briefly addressing them, we can say in addition to 3.2–3.5 above that

Word was a significant factor in the experimental design ($F[16,1530] = 11.58, p < .001$). Across all dB levels and in the critical industrial noise contexts, the following five words in particular yielded significantly positive ASR identification performances: *sidewalk*, *rudolph*, *look-out*, *jamming*, and *itchy*. They are all disyllabic and characterized by long sonorous sections (including diphthongs), extensive formant transitions, and/or consonants with high-intensity frication elements.

4 Discussion

The current study investigated the effects of different ASR systems, different enhancement strategies within these ASR systems, gender of the speaker, and the type of background noise on identification accuracy on target words, with a focus on industrial noise. Our findings contribute evidence that ASR systems, although making large improvements, continue to encounter significant challenges with word recognition when in combination with noisy environments. This is particularly true of industrial noise.

One of the largest effects is that of Noise Type which has the strongest effect on ASR performance for all tested conditions. Although the ASR systems handled the speech shaped noise comparatively well at lower dB levels (≤ 76 dB), their performances deteriorated drastically at the highest dB (79 dB). Conversely, industrial noise – especially the noise of ZOOM0254 (cutting and polishing machine; MAM72-70V) and ZOOM0272 (grinding machine; Planomat HP Blohm) – produced lower rates of identification across most levels however did not have the same detrimental degradation at the 79 dB level. This suggests that industrial noise, despite being unfavourable for ASR systems, may contain spectral-temporal properties which are easier for ASR systems to adapt or filter at higher dB levels. The fact that the noise from the milling machine allowed slightly more target word identification (Fig. 3 and Fig. 5) could actually be due to an acoustic energy gap that characterizes the noise pattern of this machine in the particularly sensitive range between 500 Hz and 3 kHz (Fig. 2). In this frequency range lie the first two formants of speech production, which are particularly relevant for the identification of consonants and vowels. In this respect, the higher ASR accuracy for ZOOM260 is empirically plausible; and, moreover, it shows that each voice interface must be individually tailored to the specific circumstances of noise, speaker, ASR system, and speech-enhancement availability. One-size-fits-all solutions seem difficult to implement in view of our results.

Surprisingly, the addition of various enhancements to each of the ASR systems did not enhance recognition. This may be due to the nature of the enhancements focusing on speech enhancements rather than noise suppression. Trials for ASR system without enhancement had better target word recognition rates than trials with enhancements, especially for the female speaker in the speech shaped noise conditions. This result contrasts with the presumed utility of the enhancements provided by metricGAN + and MTL-Mimic, suggesting that these frontend processing steps may create distortions or potentially mask relevant speech features, particularly for higher pitched speech inputs like those of female voice. Together, these results align with more frequently emerging concerns in ASR research regarding one-size-fits-all preprocessing approaches, which may generalize effectively across assorted acoustic conditions or even speaker types.

The interaction ASR type and enhancement complicates the results further. The three higher performing ASR systems (canary, parakeet, and whisper – large) benefitted

the most from the absence of enhancements versus their lower performing counterparts (wav2vec2 and whisper – small). This suggests potential incompatibilities between the current algorithms of enhancements and the noise treatment mechanisms integrated within more advanced ASR models, which may already integrate more robust speech representation techniques.

Gender effects were also consistently observed. The female voice demonstrated higher rates of identification than the male counterpart across most noise types or levels. These effects were more pronounced in industrial noise conditions without enhancements. This implies that perhaps the acoustic characteristics of female speech, such as the fundamental frequency, formant spacing, or speech clarity, may provide advantages in noisy conditions. Further investigation into the impacts of different speaker characteristics on ASR robustness may provide important information to understanding the effects demonstrated by the present research. With the female voice’s advantage, we replicate our previous finding from [14, 15]. The only notable difference is that, for the commercial ASR systems of Google and Apple, the advantage of the female voice was particularly pronounced for white noise and barely noticeable with industrial noise. In this study, i.e. with the open-source ASR systems, was it the other way around. It is important to note that the inferences made regarding gender in the present study is based only two speakers: one male, one female. This calls for future more in-depth follow-up investigations that also take into account the enhancement factor and includes a larger numbers of male and female speakers.

Lastly, the main effects of ASR type were generally weak, supporting the notion that newer or even larger scale models like whisper – large – and parakeet outperform small or older systems, particularly in challenging conditions like background noise. However, the lack of significant interactions between ASR type and noise type may suggest that structural differences in ASRs are less critical than how the system integrates and adapts to frontend processing and real-world acoustic variability.

5 Conclusions

The findings of the current study provide important insights into the complexity of interplay between the structure of ASR systems and the acoustic environment. These findings provide novel results for ASR performance in industrial noise, an underexamined area of research often focusing on, while simultaneously anchoring these results against the more well-studied noises, like white noise and speech shaped noise. Understanding this interplay is critical in the implementation of ASR into noisy real-world environments like those of factory halls.

Across all noise levels, ASR systems struggled with accuracy for target word identification with their performance significantly lower compared to our previous study using more commercially available systems of Google and Apple [14, 15]. More recent, open-source systems, despite their advancements, may lack the robustness necessary for more acoustically adverse environments without additional adaptation or training. Even at the lowest test noise level (67 dB), the recognition accuracy was low indicating the threshold for effective ASR performance in industrial noise contexts may be situated below the currently accepted benchmarks.

Note the differences in testing between the present study and our previous research on Google Translate and Apple Dictation [14, 15]. Our previous study used physical and manual manipulation of the noise signal to obtain results simulating ASRs functioning in a real acoustic environment. Although the initial baseline measurements were collected in a real acoustic environment, the present testing was preformed through traditional, digital ASR testing methods. The differences between the testing methods, in part, affect the large differences in results between the present and previous study.

Background noise type appears to be the strongest factor affecting the accuracy of target word recognition. Industrial noises imposed a greater challenge for ASR systems than speech shaped noise, especially at more moderate noise levels. Interestingly, this pattern appears reversed at the highest noise level (79 dB) where the speech shaped noise led to the steepest decline in performance. This inversion suggests a complex interaction between signal-to-noise ratios and the spectral density of competing signals. It also highlights the need for ASR models to handle noise conditions which are both predictable and dynamic in nature. The variability occurring across noise types of emphasis the inadequacy of using white noise or speech shaped noise as a proxy for real world acoustic interferences in industrial sectors like factory halls when evaluating the robustness of an ASR system.

Contradictory to the expected results, opting for ASRs without enhancements consistently led to higher identification rates for target words. This challenges the common assumption that signal enhancements will benefit ASR systems. One possible explanation may be due to the nature of the stimuli themselves which are short, disyllabic utterances. The short temporal duration may be disadvantageous, not providing enough acoustic information to assist in identification for the different types of enhancements. In their current structure, these enhancements techniques may interfere with the internal speech representations of present ASR models. This effect is especially detrimental for female voice input, where enhancements significantly reduced the accuracy of recognition. These findings highlight the need for greater alignment between strategies of enhancements and the process of speech encoding in ASR systems.

Gender effects were also notable. Female speech was more accurately recognized across most conditions, more so without enhancements. These results imply, as mentioned previously, there may be certain acoustic properties of female speech which are advantageous to ASR systems in the separations of speech from noise. This calls more inclusive training practice which are more representative of gender differences. This also calls for potential tailoring in ASR systems through the leverage of speaker specific acoustic features when utilizing ASR systems in noise-adverse environments.

With all these results taken together, several challenges arise. Firstly, general-purpose ASR systems remain susceptible to breakdowns in noisy, industrial environments. Secondly, many widely adopted enhancement methods may not provide improvements to the speech signal and, in some cases, may hinder speech recognition. Thirdly, speaker characteristics appear to play an important role in word recognition yet are often unaccounted for in model designs and evaluations. Addressing these challenges require a shift in current models and methods to more adaptable ASR pipelines which consider aspects like the acoustic paradigms of the environment, the nature of the speech input, and the interaction between enhancement and the stages of recognition. Future research

should focus on developing noise-type training, as well as gender responsiveness structures, and, most importantly, designing enhancement methods which compliment rather than compete with ASR processing. Until these integrative approaches can be defined, refined, and implemented, ASR systems remain at insufficient levels of robustness necessary and needed for real world deployment in industrial environments and perhaps may not be the most optimal in design for the current needs.

5.1 Limitations and Future Work

There are two key limitations of this research. Firstly, as preliminary work facilitating early experimentation into the viability of speech technology interfaces on factory floors, we cannot yet appropriately validate the similarity between simulated data and real-world speech data in a factory environment. It is important to note, that whilst this study is not focused on connected speech, that this will be an important aspect of how speech is produced to interface with these technologies. In addition, adaptation of speech to both industrial environments, and adaptivity to interacting with speech interfaces – an important and growing area of research – are not explored within this study but will be engaged with appropriately once non-synthetic data collection has concluded. Secondly, this study deals with the viability of using ASR systems in industrial environments as preliminary research exploring the potential for key words to be used as non-lexical items for machine control. There are approaches to limited vocabulary speech-based command systems that do not use ASR systems; however, these fall out of this scope of this initial project but are being actively considered as alternative approaches for full system deployment.

Additionally, the use of ASR in real world environments is not commonly representative of advertised Word Error Rates (WER), particularly regarding diverse speaker corpora [18–20]. Whilst one might assume ASR technologies respond well to simple single word identification there is growing research in the domain of verbal fluency transcription showing this to be a false assumption [21]. This indicates that simple limited dictionary-based speech processing models may yield superior results - in these contexts - to more generalized ASR models.

Future work may wish to explore the viability of fully computational simulation for preliminary work dealing with specific types of noise and target environments for speech technology deployment.

Acknowledgments. The authors would like to thank the European Commission and Arrowhead flexible Production Value Network (fPVN) project (ECSEL JU grant agreement No. 101111977), the Innovation Fund Denmark (2126-00004B), and the Leverhulme Trust (ECF-2025-096). AI was used to streamline some text elements.

References



1. Jurafsky, D., Martin, J.H.: *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Pearson, London (2009)

2. Lin, K.R.: Towards inclusive voice user interfaces: a systematic review of voice technology usability for users with communication disabilities. In: Stephanidis, C. et al. (eds.) *HCI International 2024*. LNCS, pp. 75–85. Springer, Cham (2024)
3. Rzepka, C.: Examining the use of voice assistants: a value-focused thinking approach. In: *Proceedings of 25th Americas Conference on Information Systems (AMCIS)*, Cancun, Mexico, pp. 1–10 (2019)
4. Junqua, J.C., Haton, J.P.: *Robustness in Automatic Speech Recognition: Fundamentals and Applications*, vol. 341. Springer, Berlin (2012)
5. Vajpai, J., Bora, A.: Industrial applications of automatic speech recognition systems. *Int. J. Eng. Res. Appl.* **6**(3), 88–95 (2016)
6. Varga, P., et al.: Making system of systems interoperable-the core components of the arrowhead framework. *J. Netw. Comput. Appl.* **81**, 85–95 (2017)
7. Rodrigues, A., Santos, R., Abreu, J., Beça, P., Almeida, P., Fernandes, S.: Analyzing the performance of ASR systems: the effects of noise, distance to the device, age and gender. In: *Proceedings of 20th International Conference Human Computer Interaction*, pp. 1–8 (2019)
8. Lee, S.H., Shim, H.J., Yoon, S.W., Lee, K.W.: Effects of various background noises on speech intelligibility of normal hearing subjects. *Korean J. Otorhinolaryngol.-Head Neck Surg.* **52**(4), 307–311 (2009)
9. Shukla, B., Rao, B.S., Saxena, U., Verma, H.: Measurement of speech in noise abilities in laboratory and real-world noise. *Indian J. Otol.* **24**(2), 109–113 (2018)
10. Summers, W.V., Pisoni, D.B., Bernacki, R.H., Pedlow, R.I., Stokes, M.A.: Effects of noise on speech production: acoustic and perceptual analyses. *J. Acoust. Soc. Am.* **84**(3), 917–928 (1988)
11. Lu, Y., Cooke, M.: Speech production modifications produced by competing talkers, babble, and stationary noise. *J. Acoust. Soc. Am.* **124**(5), 3261–3275 (2008)
12. Brungart, D.S.: Informational and energetic masking effects in the perception of two simultaneous talkers. *J. Acoust. Soc. Am.* **109**(3), 1101–1109 (2001)
13. Fontan, L., Tardieu, J., Gaillard, P., Woisard, V., Ruiz, R.: Relationship between speech intelligibility and speech comprehension in babble noise. *J. Speech Lang. Hear. Res.* **58**(3), 977–986 (2015)
14. Pearsell, S.M., Niebuhr, O.: Command and conquer: towards a robust inventory of voice commands for HMI in factory halls. In: Dau, T., Epp, B. (eds.) *Proceedings of DASIDAGA 2025 – 51st Annual Meeting on Acoustics*, pp. 740–743. DEGA, Berlin (2025). https://pub.dega-akustik.de/DAS-DAGA_2025
15. Pearsell, S.M., Niebuhr, O.: Lost in the noise: evaluating ASR performance in industrial and environmental noise. In: *Proceedings of 2025 8th IEEE International Conference Industrial Cyber-Physical System (ICPS)*. IEEE, Emden (2025)
16. Hellman, R., Zwicker, E.: Why can a decrease in dB(A) produce an increase in loudness? *J. Acoust. Soc. Am.* **82**(5), 1700–1705 (1987)
17. Boersma, P., Weenink, D.: Praat: Doing Phonetics by Computer (Version 6.4.37) [Computer Program]. Accessed 15 May 2025. <http://www.praat.org/>
18. Schmück, S., Blackburn, D., Christensen, H.: Evaluating the performance of state-of-the-art ASR systems on non-native English using corpora with extensive language background variation. In: *Proceedings of Interspeech 2022 – Annual Conference on International Speech Communication Association*, pp. 3958–3962. ISCA, Incheon (2022)
19. Koenecke, A., et al.: Racial disparities in automated speech recognition. *Proc. Natl. Acad. Sci. U.S.A.* **117**(14), 7684–7689 (2020). <https://doi.org/10.1073/pnas.1915768117>
20. Gerosa, M., Giuliani, D., Narayanan, S., Potamianos, A.: A review of ASR technologies for children’s speech. In: *Proceedings of 2nd Workshop on Child, Computer and Interaction*, pp. 1–8 (2009)

21. Pakhomov, S.V., Marino, S.E., Banks, S., Bernick, C.: Using automatic speech recognition to assess spoken responses to cognitive tests of semantic verbal fluency. *Speech Commun.* **75**, 14–26 (2015). <https://doi.org/10.1016/j.specom.2015.09.010>



Ground Truth-Free WER Prediction for ASR via Audio Quality and Model Confidence Features

Anton Polevoi¹, Alexander Kragin², and Natalia Loukachevitch³

¹ Lomonosov Moscow State University, Moscow, Russia
polevoianton@bk.ru

² AI Talent Hub, ITMO University, St. Petersburg, Russia

³ Research Computing Center, Lomonosov Moscow State University, Moscow, Russia

Abstract. We propose a data-driven approach for predicting Word Error Rate (WER) without requiring ground truth transcriptions. Our method involves creating diverse audio datasets by applying various noise types, acoustic degradations, and room impulse responses to clean speech samples across many fine-grained quality and intelligibility levels. Unlike previous work, we extract and analyze a comprehensive set of speech quality features including signal-to-noise ratio (SNR) estimates, modern neural audio quality metrics (such as NISQA), and ASR (Automatic Speech Recognition) model confidence scores to train WER prediction models. We conduct experiments across multiple languages with state-of-the-art ASR architectures (Whisper and FastConformer) to demonstrate our method's effectiveness in predicting WER in diverse acoustic conditions. We also show that our approach generalizes in a multilingual model-unified setting. We provide feature importance analysis to identify key metrics needed to predict WER. This work enables practical applications such as quality-based filtering of audio inputs, allowing ASR systems to assess expected performance and estimate transcription reliability without ground truth transcripts.

Keywords: ASR · Word error rate prediction · Audio augmentation dataset

1 Introduction

Deep neural networks have demonstrated remarkable performance in controlled settings, yet their robustness in real-world applications—particularly in automatic speech recognition (ASR)—remains a persistent challenge. A critical gap exists in diagnosing ASR failures before deployment, especially when acoustic degradations or linguistic complexity induce errors ranging from minor inaccuracies to catastrophic hallucinations. Traditional evaluation relies on post-hoc transcription analysis ([37]), which is impractical for real-time systems. Instead, we argue that predicting Word Error Rate (WER) from simulated degradations

A. Polevoi and A. Kragin—Equal contribution.

offers a proactive solution: by synthetically replicating real-world distortions (e.g., noise, reverberation, or packet loss), we can directly correlate degradation severity with ASR failure rates, where higher predicted WER signals imminent model unreliability.

The primary causes of faulty transcriptions fall into two categories:

- **Acoustic Degradations:** Environmental noise, microphone artifacts, reverberation, and signal distortion—all synthetically simulatable to stress-test ASRs.
- **Speech Complexity:** Accents, rare words, and domain-specific terms, which require curated datasets to evaluate robustness.

Crucially, while linguistic challenges are harder to simulate, acoustic degradations can be precisely controlled and reintroduced into clean audio. This enables systematic analysis of how specific distortion types (e.g., additive noise vs. clipping) and their intensity (e.g., SNR levels) degrade ASR performance. By modeling the relationship between these degradations and WER, we can preemptively flag audio inputs likely to cause failures, even without ground-truth transcriptions.

Our work is based on the core theses:

- **Simulated degradations expose ASR vulnerabilities.** By artificially degrading clean audio with real-world distortions (e.g., dynamic noise profiles or impulsive interruptions), we create a controlled environment to quantify how specific degradation types and levels correlate with WER elevation—a direct proxy for model failure.
- **WER prediction is feasible without transcripts.** Using two feature groups—(1) ASR Model Confidence Metrics, (2) Audio Quality Features (WADA SNR, SI-SNR, NISQA), we demonstrate that WER can be accurately predicted, enabling real-time diagnostics.

Our contributions include:

- A framework for simulating diverse real-world degradations (e.g., variable SNR noise, reverberation tails, codec artifacts) and evaluating their impact on state-of-the-art ASR models (Whisper, FastConformer).
- A regression model to predict WER solely from Speech Quality Features and ASR Model Confidence Metrics.

2 Background

2.1 Acoustic Degradations for ASR Model

Existing research on ASR robustness had primarily focused on defending models against acoustic degradations. According to Shah et al. [2], these approaches can be categorized as either model-based or feature-based.

Model-based approaches include adapting pre-trained models [4, 5], pre-processing audio with denoising techniques [6, 7], and training directly on noisy

data [8]. These strategies typically require access to representative noisy data and are most effective when the deployment environment and noise characteristics are known in advance.

Feature-based approaches focus on developing noise-invariant representations of speech. These include biologically-inspired features [9] and signal processing techniques [1] designed to extract speech-relevant components while filtering out irrelevant signal elements [10].

In recent years it has been demonstrated that large-scale training on diverse acoustic scenarios can produce inherently robust ASR systems without specialized techniques. Notably, Radford et al. [3] showed that models like Whisper achieve substantial robustness through exposure to varied audio conditions during training.

Typically, the performance of a new ASR system is evaluated on multiple standard speech datasets, which inherently contain varying levels of noise and signal quality. This approach by itself provides some insight into the system’s robustness. However, a more direct method to assess robustness across different conditions involves simulating various signal degradations and measuring the Word Error Rate (WER) at different Signal-to-Noise Ratio (SNR) levels, as demonstrated in numerous studies on ASR robustness [3, 29–31].

2.2 Word Error Rate (WER)

Word Error Rate (WER) is the standard evaluation metric for automatic speech recognition systems. It measures the distance between a reference transcript and the ASR model’s hypothesis by calculating the minimum number of word-level operations (substitutions, insertions, and deletions) required to transform the hypothesis into the reference.

WER is defined as:

$$\text{WER} = \frac{S + D + I}{N} \quad (1)$$

where S is the number of substituted words, D is the number of deleted words, I is the number of inserted words, N is the total number of words in the reference transcript. Lower WER values indicate better ASR performance, with 0.0 representing perfect transcription. WER can exceed 1.0 in cases where the number of insertion errors is particularly high.

2.3 WER Prediction

Because WER serves as an effective indicator of ASR performance, the main motivation for developing WER prediction methods is to assess the quality of ASR predictions without ground-truth transcriptions, which are often unavailable in real-world applications.

Several methods have been proposed to predict WER without ground truth transcriptions. For instance, eWER [11] detects disagreement between 2 ASR systems (word-level and character-level) and uses that to predict WER. eWER-2

[12] utilizes acoustic, lexical, and phonotactic features for the same task. eWER-3 [13] demonstrated that their multilingual model outperforms previous monolingual WER prediction methods (eWER2) by achieving a 9% absolute increase in Pearson correlation coefficient (PCC), showing better correlation between predicted and reference WER. A common downside of these methods is the need for either 2 separate ASR models or additional models to process text and extract phonemes, which increases computational requirements and overall complexity.

Other approaches include Litman et al. [14], who calculated prosodic statistics (frequencies, energy values, tempo, pauses) and used an ML model to predict WER for telephone dialogue recordings. Fish et al. [15] estimated SNR and decomposed WER into base WER (from inherent model limitations) and delta-WER (from audio quality degradations). Additionally, Gallardo et al. [16] investigated speech quality measurement techniques like POLQA [19] to predict WER. Their results led to polynomial models for predicting speech recognition accuracy from instrumental measures across various channel distortions in different bandwidths.

Some of the most recent works by Park et al. [20,21] achieve impressive prediction accuracy, but rely on using natural language processing techniques to incorporate information from ASR hypothesis transcriptions.

To summarise, our approach differs in several key ways:

- We predict WER using only audio quality features and confidence scores, avoiding extra complexity associated with using additional natural language processing and/or phone recognition models.
- Our training and evaluation pipelines both use speech samples in different languages under diverse acoustic conditions, including different types of noise and various signal degradations.
- We conduct experiments with the most common modern ASR architectures, namely Whisper [3] and FastConformer [27].
- Our approach relies on a lightweight WER regressor model combined with a modular feature extraction architecture. This design ensures flexibility, allowing researchers to incorporate new speech quality estimation methods as they emerge.
- We provide feature importance comparisons to facilitate further research into ASR WER prediction and speech quality metrics.

3 Proposed Solution

3.1 Degradation Framework

To train a robust WER prediction model that performs reliably across diverse acoustic conditions, we need a comprehensive dataset. We create this dataset by creating speech recordings that span a wide range of scenarios. Starting with clean speech recordings, we systematically add different types of noise at varying intensity levels and apply additional augmentations (aliasing, real impulse responses, MP3 compression, room simulation), as detailed in Algorithm 1. This

synthetic data generation strategy allows us to produce a large, diverse dataset with known ground truth transcriptions and controlled degradations, enabling our WER prediction model to learn patterns of ASR performance across a broad spectrum of real-world acoustic environments.

We design the data generation pipeline as a highly modular system to support adding arbitrary ASR models and audio feature extractors, both local and available through external APIs.

Algorithm 1. Dataset Creation

Require: Clean speech dataset with transcriptions, noise files, gain levels, ASR transcriber, feature extractors

Ensure: Dataset of features paired with ground truth WER values

```

1: Filter dataset to desired duration range
2: Sample subset of audio files from filtered dataset
3: Initialize results collection
4: for each audio file with transcript do
5:   Load and normalize audio
6:   for each noise gain level do
7:     Initialize record with metadata
8:     Select random noise and extract segment matching audio length
9:     Scale noise by gain factor and mix with clean audio
10:    Apply normalization if needed
11:    Apply augmentations
12:    Generate ASR transcription
13:    Compute ASR confidence scores from model logits
14:    Calculate SNR and WER
15:    Extract all quality features using feature extractors
16:  end for
17: end for
18: Save raw results
19: Compute aggregated statistics by gain level
20: return Complete dataset

```

3.2 Speech Quality Feature Extractors

To obtain audio quality and speech intelligibility estimates we leverage 3 distinct speech quality assesment methods as feature extractors for the WER prediction model:

1. WADA-SNR (Waveform Amplitude Distribution Analysis) [23]: a classical statistical approach that estimates signal-to-noise ratio by modeling clean speech as a Gamma distribution and noise as a Gaussian distribution.
2. SpeechBrain SI-SNR Estimator [22]: a modern neural approach that blindly estimates Scale-Invariant Signal-to-Noise Ratio.

3. NISQA (Non-Intrusive Speech Quality Assessment) [24]: a modern neural approach to provide multidimensional audio quality assessment. NISQA evaluates not only overall quality but also specific dimensions (noisiness, coloration, discontinuity, and loudness).

These feature extractors are selected for speed, ease of use and the ability to capture different aspects of audio quality.

3.3 ASR Model Confidence Metrics

To quantify ASR model uncertainty, we utilise a variety of confidence metrics derived from the model’s token-level or char-level log probabilities (log-probs). These confidence metrics are used as features to provide signals about the model’s certainty in its predictions. We implement and evaluate multiple confidence formulations to determine which best correlate with transcription accuracy:

- **Least Confidence:** Normalizes the uncertainty represented by the difference between the maximum probability and 1.0.

$$LC = \frac{1 - \max_i(p_i)}{1 - \frac{1}{|\mathcal{V}|}} \times \frac{|\mathcal{V}|}{|\mathcal{V}| - 1} \quad (2)$$

- **Margin Confidence:** Measures the difference between the probabilities of the top two predictions.

$$MC = 1 - (p_{(1)} - p_{(2)}) \quad (3)$$

- **Ratio Confidence:** Calculates the ratio between the second highest and highest probabilities.

$$RC = \frac{p_{(2)}}{p_{(1)} + \epsilon} \quad (4)$$

- **Entropy-Based Confidence:** Uses normalized Shannon entropy of the probability distribution.

$$EC = -\frac{1}{\log_2(|\mathcal{V}|)} \sum_{i=1}^{|\mathcal{V}|} p_i \log_2(p_i + \epsilon) \quad (5)$$

- **Perplexity Confidence:** Applies normalized perplexity measure.

$$PC = \frac{\exp(-\sum_{i=1}^{|\mathcal{V}|} p_i \log p_i) - 1}{|\mathcal{V}| - 1} \quad (6)$$

- **Gini Coefficient Confidence:** Adapts the Gini coefficient to measure prediction inequality.

$$GC = \frac{2 \sum_{i=1}^{|\mathcal{V}|} i \cdot p_{(i)}}{|\mathcal{V}| \sum_{i=1}^{|\mathcal{V}|} p_{(i)}} - \frac{|\mathcal{V}| + 1}{|\mathcal{V}|} \quad (7)$$

- **Mean Chosen Confidence:** Takes the mean of maximum logprobs across tokens.

$$\text{MCC} = \frac{1}{n} \sum_{j=1}^n \max_i (\log p_{i,j}) \quad (8)$$

Where p_i represents the probability of token i , $p_{(i)}$ denotes the i -th largest probability in the distribution, $|\mathcal{V}|$ is the vocabulary size, ϵ is a small constant to prevent division by zero, and n is the number of tokens in the sequence. We calculate these metrics for each token in the ASR output and average them to obtain utterance-level confidence scores.

These confidence metrics capture different aspects of prediction uncertainty: entropy-based measures assess overall distribution spread, margin and ratio confidence focus on ambiguity between top predictions, while Gini coefficient measures the inequality of the probability mass.

3.4 WER Prediction Model

We propose a robust WER prediction framework that utilises ASR model confidence scores as well as the speech quality feature extractors listed above to account for various audio degradations. These features serve as inputs to a regression model. We evaluate several approaches: classical (Linear Regression, Random Forest, CatBoost [35]), AutoML Ensembles (LightAutoML [36]) and modern DL tabular methods (TabPFN [33,34]). TabPFN with vanilla parameters was used for the final experiments.

4 Experiments

Data. We run our experiments on 2 distinct datasets to account for differences in data and explore a multilingual setting.

- 1,000 randomly selected recordings from Fleurs [17] (Russian)
- 1,000 randomly selected recordings from LibriSpeech [18] (English)

First, we randomly split the original recordings into train and test sets, then apply degradations separately within each set to avoid mixing augmented versions across splits.

Noise Application. We implement a linear noising schedule with 64 discrete steps, ranging from SNR +60 dB (clean audio) to SNR -10 dB (extremely degraded audio that is challenging or impossible for humans to transcribe). For each recording, we randomly apply one of the following noise types:

1. Bank-easy: Background sounds from a bank environment with minimal background speech
2. Office-easy: Background sounds from an office environment with minimal background speech
3. Office-hard: Background sounds from an office environment containing audible background speech
4. White noise: Standard white noise distortion

Audio Augmentation. In addition to controlled noise addition, we apply various audio augmentations using the Audiomentations library [32]. Our augmentation pipeline includes aliasing effects (50% probability), impulse response convolution (30% probability) using responses from the MIT IR Survey [28], MP3 compression artifacts (50% probability), and room simulation effects (70% probability). These augmentations introduce realistic audio degradations commonly encountered in real-world scenarios.

ASR Models. Our experiments utilize a diverse set of ASR models to represent modern speech recognition approaches across different parameter scales:

- Whisper [3] variants: Turbo (800M) and Small (240M)
- FastConformer [27] with CTC [25] Decoder (120M)

For inference, we use the original OpenAI Whisper implementation for the Whisper variants and Nvidia NEMO Toolkit [26] for FastConformer.

All models are evaluated in both English and Russian languages to assess cross-lingual robustness. This selection is meant to encompass modern SOTA ASR approaches across different model sizes.

4.1 Adequacy of Proposed Features

SNR Estimation in the Presence of Audio Augmentations. Our experiments provide insights into how different SNR estimation methods perform under varying audio conditions, as illustrated in Fig. 1. We find that WADA SNR, which relies on simple statistical assumptions about the distribution of speech and noise, performs admirably under standard noising conditions (correlation of 0.8983). However, its performance deteriorates substantially (dropping to 0.6246) when confronted with complex audio augmentations such as MP3 compression artifacts, impulse response convolution, and room simulations.

In contrast, more sophisticated modern neural-based techniques like Speech-Brain SI-SNR models appear to be more resilient. This suggests that modern neural approaches to SNR estimation have implicitly learned more robust representations that can better generalize across various audio degradation types beyond simple additive noise.

Correlation Between Speech Quality Features, ASR Model Confidence Metrics and WER. We observe that ASR Model Confidence Metrics exhibit the strongest correlation with WER, generally achieving correlation coefficients of approximately 0.8, as shown in Table 1. Despite our intuitive expectations that Speech Quality Features would serve as the most reliable predictors of ASR performance, the model’s own confidence score proved to be better indicators of WER.

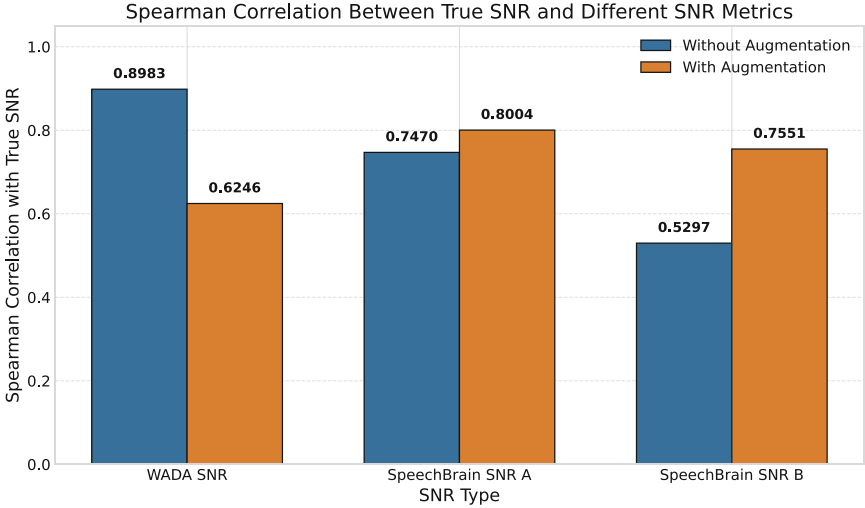


Fig. 1. Spearman correlation between true SNR and different SNR estimation techniques, comparing performance with and without audio augmentations. WADA SNR shows significant performance degradation with augmentations, while SpeechBrain model demonstrates greater robustness.

4.2 Results

WER Prediction Performance Across Models and Languages. Our WER prediction models demonstrate strong performance across different ASR architectures and languages, as shown in Fig. 2. There are several notable patterns:

First, we observe that WER prediction accuracy varies by language, with consistently better performance on English data across all ASR models. This is likely attributable to the prevalence of English in training data for both ASR systems and audio quality assessment models, resulting in more reliable feature extraction and confidence scoring. For instance, Whisper Turbo achieves an RMSE of 0.0633 on LibriSpeech (English) compared to 0.0768 on Fleurs (Russian).

Second, our multilingual approach combining Russian and English data into a single prediction model maintains strong performance despite the challenges of cross-language generalization. As illustrated in Fig. 2, the combined multilingual model achieves competitive RMSE values (0.0747 for Whisper Turbo, 0.0943 for Whisper Small, and 0.1060 for FastConformer), demonstrating the feasibility of language-agnostic WER prediction. This suggests that the fundamental relationship between audio quality metrics, confidence scores, and transcription accuracy generalizes across languages, at least for the high-resource languages tested.

Table 1. Correlation coefficients between various features and WER. ASR confidence metrics consistently show higher correlation (approximately 0.8) compared to speech quality metrics.

Correlation Group	Feature	Correlation Coefficient (r)
High ($ r \geq 0.7$)	Entropy Confidence	0.82
	Variance Confidence	-0.81
	Least Confidence	0.81
	Temperature Scaled Confidence	0.80
	Margin Confidence	0.79
	Jensen Shannon Confidence	-0.78
	Mean Chosen Confidence	-0.78
	Top K Entropy Confidence	0.76
Mid ($0.2 \leq r < 0.7$)	Gini Confidence	-0.69
	Ratio Confidence	0.69
	NISQA MOS	-0.56
	SpeechBrain SNR A	-0.51
	Perplexity Confidence	0.50
	NISQA Discontinuity	0.28
	SpeechBrain SNR B	-0.26
	NISQA Noisiness	-0.25
	NISQA Coloration	-0.25
Low ($ r < 0.2$)	NISQA Average	-0.19
	WADA SNR	-0.09
	NISQA Loudness	0.05

Third, among the ASR architectures evaluated, Whisper Turbo consistently outperforms smaller models in WER prediction accuracy across all language settings. With an RMSE of 0.0747 on the combined multilingual dataset, Whisper Turbo demonstrates a 20.8% relative improvement over Whisper Small (0.0943) and a 29.5% improvement over FastConformer (0.1060). This performance gap suggests that larger, more capable ASR models not only transcribe more accurately but also provide more reliable confidence scores that correlate better with actual transcription quality.

A detailed breakdown of the metrics including correlation coefficients is available in Table 2.

Table 2. RMSE and correlation results when predicting WER for a particular ASR model.

Model	Size	LibriSpeech (en) Only			Fleurs (ru) Only			Combined Multilingual		
		RMSE	Pearson	Spearman	RMSE	Pearson	Spearman	RMSE	Pearson	Spearman
whisper-large-v3-turbo	809M	0.0633	0.9891	0.9424	0.0768	0.9827	0.9145	0.0747	0.9847	0.9179
whisper-small	244M	0.0752	0.9817	0.9545	0.1077	0.9527	0.9399	0.0943	0.9690	0.9541
fast-conformer	120M	0.0758	0.9828	0.9340	0.0866	0.9682	0.8942	0.1060	0.9604	0.9168

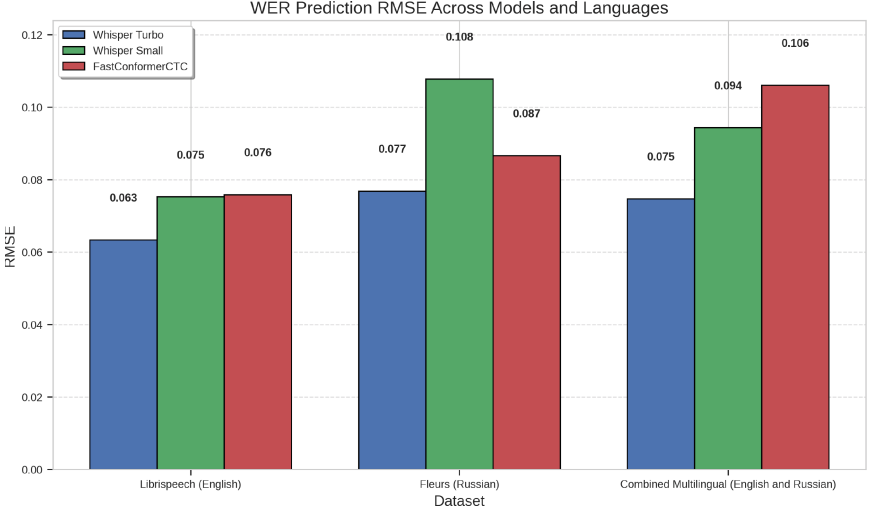


Fig. 2. RMSE for WER prediction across different models and datasets. Lower values indicate better prediction accuracy. The multilingual dataset combines data from Russian and English datasets.

Model-Unified Multilingual WER Prediction. While model-specific WER predictors demonstrate superior performance, maintaining separate predictors for each ASR model and language combination quickly becomes impractical in real-world deployments. To address this challenge, we investigate a model-unified approach that enables WER prediction across different ASR architectures and languages using a single prediction model. This approach offers significant practical advantages: it eliminates the need for continuous retraining as new ASR models are developed, provides a unified quality assessment framework across multilingual applications, and simplifies system architecture. Furthermore, as speech recognition technology evolves, a generalizable predictor that works reasonably well with new ASR models without immediate retraining provides a more sustainable and scalable solution for real-world production deployment.

Our model-unified approach involves training a regression model on the combined data from all ASR systems (Whisper Turbo, Whisper Small, and FastConformer) across both English and Russian languages. This unified predictor relies on the same feature set described earlier, including audio quality metrics and ASR confidence scores, but learns to generalize across different ASR models and architectures.

As shown in Fig. 3, the model-unified approach achieves an RMSE of 0.0994 on the combined multilingual dataset. While this performance is slightly worse than the best model-specific predictor (Whisper Turbo at 0.0747, representing a 24.9% relative difference), it outperforms the FastConformer-specific predictor (0.1060) by 6.2%. This indicates that a single prediction model can estimate

WER reasonably well across different ASR architectures without requiring separate predictors for each system.

The effectiveness of this approach suggests that despite architectural differences between ASR models, there exist common patterns in how speech quality and model confidence scores correlate with transcription accuracy. As new ASR models are developed, existing model-unified WER predictors may still provide reasonable estimation accuracy without requiring immediate retraining, although periodic fine-tuning with data from newer models would likely improve performance over time.

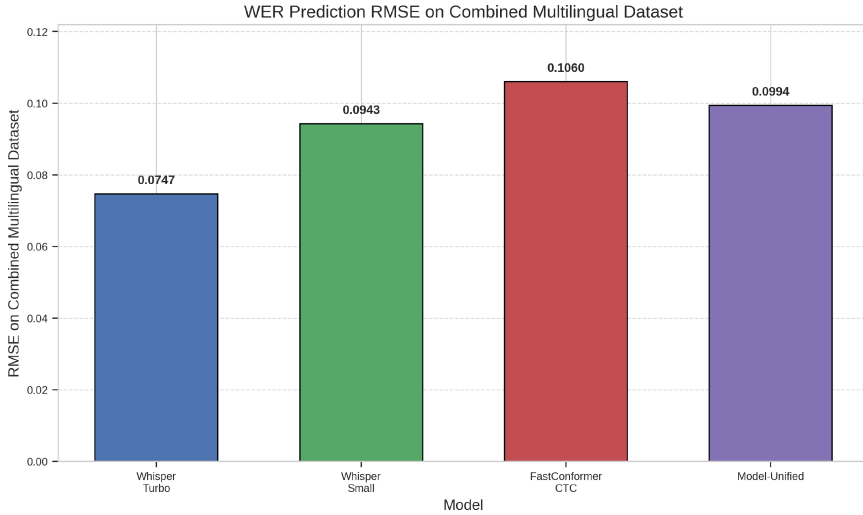


Fig. 3. RMSE for WER prediction for the combined multilingual dataset. Lower values indicate better prediction accuracy. The multilingual dataset combines data from Russian and English datasets. Model-unified approach involves training a single predictor for all 3 ASR models.

5 Ablation Studies

To understand the relative importance of different feature groups in our WER prediction framework, we conducted ablation experiments by systematically removing feature categories and evaluating the impact on prediction accuracy. Figure 4 shows the results of these tests for Whisper Turbo on the LibriSpeech (English) dataset.

Our ablation results reveal that confidence metrics are the most important features for accurate WER prediction. Removing them causes the RMSE to increase dramatically from 0.0633 to 0.1609, representing a 154.2% relative degradation. This result aligns with our correlation analysis in Fig. 1, which showed confidence scores having the strongest relationship with WER.

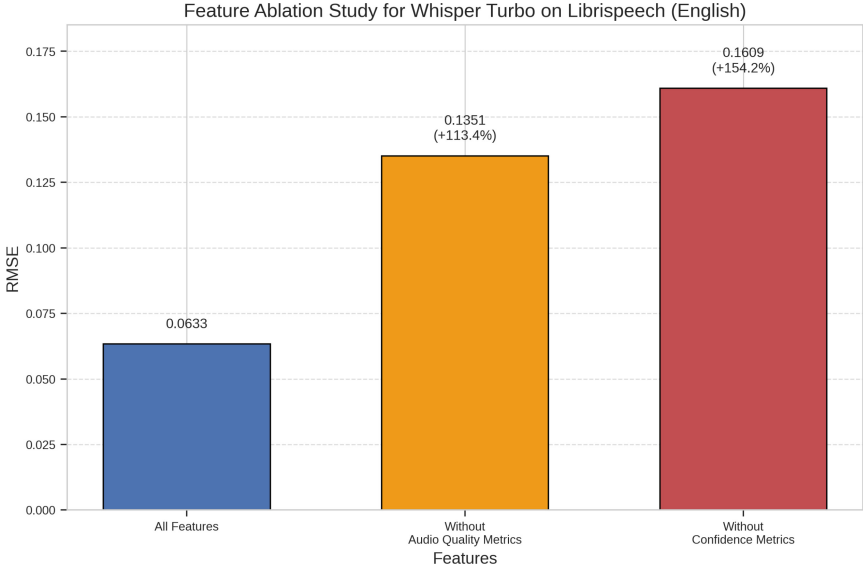


Fig. 4. Feature ablation experiments showing the impact of removing different feature categories on WER prediction performance. Percentages indicate relative RMSE increase compared to using all features.

Audio quality metrics also contribute substantially to prediction accuracy, though to a lesser extent than confidence features. Without audio quality metrics, the RMSE increases to 0.1351, a 113.4% relative increase. This shows that while confidence scores provide the strongest predictive signal, audio quality metrics also capture important information that significantly improves overall prediction accuracy.

Interestingly, even without confidence metrics (which require running ASR inference), the model still achieves reasonable prediction performance using only audio quality features. In scenarios where computational resources are limited or where a quick quality assessment is needed before deciding whether to run a full ASR pipeline, audio quality metrics alone can provide a useful rough approximation of expected transcription accuracy. For applications such as filtering out low-quality audio that would likely result in unreliable transcriptions, this approach offers a computationally efficient pre-screening mechanism without requiring full ASR inference.

6 Conclusion

This paper presents a robust approach for WER prediction without requiring ground truth transcriptions. Our key contributions include:

First, we developed a synthetic data generation pipeline that creates diverse acoustic scenarios ranging from clean to heavily degraded speech. This approach

enables comprehensive evaluation of ASR robustness across controlled degradation conditions.

Second, we demonstrated that a combination of speech quality metrics and ASR confidence scores can effectively predict WER across different ASR architectures and languages. Our analysis reveals that while modern neural audio quality metrics show promise, ASR confidence scores remain the strongest predictors of transcription accuracy.

Third, we showed that multilingual WER prediction is feasible, with our combined English-Russian model achieving strong performance across languages. Furthermore, our model-unified approach enables WER prediction across different ASR architectures using a single prediction system, simplifying deployment in real-world applications.

Our findings have significant implications for practical ASR deployments, enabling systems to estimate transcription reliability without ground truth, facilitating quality-based filtering, and improving overall robustness in diverse acoustic environments. In future work we intend to explore and integrate additional features to further improve prediction accuracy.

References

1. Li, J., Deng, L., Gong, Y., Haeb-Umbach, R.: An overview of noise-robust automatic speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **22**(4), 745–777 (2014)
2. Shah, M.A., Noguero, D.S., Heikkilä, M.A., Raj, B., Kourtellis, N.: Speech robust bench: a robustness benchmark for speech recognition. *arXiv preprint arXiv:2403.07937* (2024)
3. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: *International Conference on Machine Learning*, pp. 28492–28518. PMLR (2023)
4. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **33**, 12449–12460 (2020)
5. Hsu, W.N., Bolte, B., Tsai, Y.H., Lakhota, K., Salakhutdinov, R., Mohamed, A.: Hubert: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 3451–3460 (2021)
6. Tsilfidis, A., Mporas, I., Mourjopoulos, J., Fakotakis, N.: Automatic speech recognition performance in different room acoustic environments with and without dereverberation preprocessing. *Comput. Speech Lang.* **27**(1), 380–395 (2013)
7. Loweimi, E., Ahadi, S.M., Drugman, T., Loveymi, S.: On the importance of pre-emphasis and window shape in phase-based speech recognition. In: Drugman, T., Dutoit, T. (eds.) *NOLISP 2013. LNCS (LNAI)*, vol. 7911, pp. 160–167. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38847-7_21
8. Yin, S., et al.: Noisy training for deep neural networks in speech recognition. *EURASIP J. Audio Speech Music Process.* **2015**(1), 1–14 (2015). <https://doi.org/10.1186/s13636-014-0047-0>

9. Kim, C., Stern, R.M.: Power-normalized cepstral coefficients (PNCC) for robust speech recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **24**(7), 1315–1329 (2016)
10. Stern, R.M., Morgan, N.: Hearing is believing: biologically inspired methods for robust automatic speech recognition. *IEEE Signal Process. Mag.* **29**(6), 34–43 (2012)
11. Ali, A., Renals, S.: Word error rate estimation for speech recognition: e-WER. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 2: Short Papers, pp. 20–24. *ACL* (2018)
12. Ali, A., Renals, S.: Word error rate estimation without asr output: E-wer2. *arXiv preprint [arXiv:2008.03403](https://arxiv.org/abs/2008.03403)* (2020)
13. Chowdhury, S.A., Ali, A.: Multilingual word error rate estimation: e-WER3. In: *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 1–5. *IEEE* (2023)
14. Litman, D., Hirschberg, J., Swerts, M.: Predicting automatic speech recognition performance using prosodic cues. In: *1st Meeting of the North American Chapter of the Association for Computational Linguistics*, pp. 1–8. *ACL* (2000)
15. Fish, R., Hu, Q., Boykin, S.: Using audio quality to predict word error rate in an automatic speech recognition system. Unpublished technical report (2006)
16. Gallardo, L.F., Möller, S., Beerends, J.: Predicting automatic speech recognition performance over communication channels from instrumental speech quality and intelligibility scores. In: *INTERSPEECH 2017*, pp. 2939–2943. *ISCA* (2017)
17. Conneau, A., et al.: FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech. *arXiv preprint [arXiv:2205.12446](https://arxiv.org/abs/2205.12446)* (2022)
18. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an ASR corpus based on public domain audio books. In: *ICASSP 2015*, pp. 5206–5210. *IEEE* (2015)
19. Beerends, J., et al.: Perceptual objective listening quality assessment (POLQA), the third generation ITU-T standard for end-to-end speech quality measurement part I-temporal alignment. *AES: J. Audio Eng. Soc.* **61**(6), 366–384 (2013)
20. Park, C., Lu, C., Chen, M., Hain, T.: Fast Word Error Rate Estimation Using Self-Supervised Representations for Speech and Text. *arXiv preprint [arXiv:2310.08225](https://arxiv.org/abs/2310.08225)* (2025)
21. Park, C., Chen, M., Hain, T.: Automatic Speech Recognition System-Independent Word Error Rate Estimation. *arXiv preprint [arXiv:2404.16743](https://arxiv.org/abs/2404.16743)* (2024)
22. Subakan, C., Ravanelli, M., Cornell, S., Grondin, F.: REAL-M: Towards Speech Separation on Real Mixtures. *arXiv preprint [arXiv:2110.10812](https://arxiv.org/abs/2110.10812)* (2021)
23. Kim, C., Stern, R.: Robust signal-to-noise ratio estimation based on waveform amplitude distribution analysis. In: *INTERSPEECH 2008*, pp. 2598–2601. *ISCA* (2008). <https://doi.org/10.21437/Interspeech.2008-644>
24. Mittag, G., Naderi, B., Chehadi, A., Möller, S.: NISQA: a deep CNN-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In: *INTERSPEECH 2021*, pp. 1–5. *ISCA* (2021). <https://doi.org/10.21437/Interspeech.2021-299>
25. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: *ICML 2006*, pp. 369–376. *ACM* (2006). <https://doi.org/10.1145/1143844.1143891>
26. Harper, E., Majumdar, S., Kuchaiev, O., et al.: NeMo: a toolkit for Conversational AI and Large Language Models. *NVIDIA* (2019). <https://nvidia.github.io/NeMo/>

27. Rekish, D., Koluguri, N.R., Kriman, S., et al.: Fast Conformer with Linearly Scalable Attention for Efficient Speech Recognition. arXiv preprint [arXiv:2305.05084](https://arxiv.org/abs/2305.05084) (2023)
28. Traer, J., McDermott, J.H.: Statistics of natural reverberation enable perceptual separation of sound and space. *Proc. Natl. Acad. Sci.* **113**(48), E7856–E7865 (2016). <https://doi.org/10.1073/pnas.1612524113>
29. Bouchakour, L., Debyeche, M.: Noise-robust speech recognition in mobile network based on convolution neural networks. *Int. J. Speech Technol.* **25**(1), 269–277 (2021). <https://doi.org/10.1007/s10772-021-09950-9>
30. Duarte, J., Colcher, S.: Noise-robust automatic speech recognition: a case study for communication interference. *J. Interact. Syst.* **15**(1), 670–681 (2024). <https://doi.org/10.5753/jis.2024.4267>
31. Chen, G., O’Shaughnessy, D., Tolba, H.: A performance investigation of noisy voice recognition over IP telephony networks. In: INTERSPEECH 2005, pp. 2681–2684. ISCA (2005). <https://doi.org/10.21437/Interspeech.2005-259>
32. Jordal, I., Tamazian, A., Dhyani, T., et al.: iver56/audiomentations: v0.39.0. Zenodo (2025). <https://doi.org/10.5281/zenodo.14856562>
33. Hollmann, N., Müller, S., Purucker, L., et al.: Accurate predictions on small data with a tabular foundation model. *Nature* **625**(7993), 1–9 (2025). <https://doi.org/10.1038/s41586-024-08328-6>
34. Hollmann, N., Müller, S., Eggenberger, K., Hutter, F.: TabPFN: a transformer that solves small tabular classification problems in a second. In: ICLR 2023 (2023)
35. Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A.: CatBoost: unbiased boosting with categorical features. arXiv preprint [arXiv:1706.09516](https://arxiv.org/abs/1706.09516) (2019)
36. Vakhrushev, A., Ryzhkov, A., Savchenko, M., Simakov, D., Damdinov, R., Tuzhilin, A.: LightAutoML: AutoML Solution for a Large Financial Services Ecosystem. arXiv preprint [arXiv:2109.01528](https://arxiv.org/abs/2109.01528) (2022)
37. Koenecke, A., et al.: Racial disparities in automated speech recognition. *PNAS* **117**(14), 7684–7689 (2020). <https://doi.org/10.1073/pnas.1915768117>



Enhancing Speech Recognition Through Text-to-Speech and Voice Conversion Augmentation

Yunus Emre Ozkose^(✉) and Ali Haznedaroglu

Sestek, Istanbul, Turkey

{yunusemre.ozkose, ali.haznedaroglu}@sestek.com

Abstract. Given the challenges associated with obtaining large volumes of real-world audio, the advancement of Automatic Speech Recognition (ASR) systems increasingly depends on synthetic data. This study focuses on the call center-banking domain, providing a targeted analysis of ASR models in this context. We evaluate real and synthetic datasets, using Word Error Rate (WER) and Character Error Rate (CER) as metrics, and employ a speech quality model to assess the impact on ASR performance. Our research compares Text-to-Speech (TTS) and advanced voice conversion methods, including KNNVC, Seed-VC, and Vec2Wav2, revealing significant improvements in speech quality and ASR accuracy, particularly with Seed-VC. Additionally, our domain-specific experimentation provides insight into the unique challenges and opportunities that arise when applying ASR technologies to industry-relevant settings. This highlights the potential of voice conversion technologies to enhance ASR systems, guiding future research in diverse linguistic scenarios, and paving the way for the broader application of ASR innovations across various fields.

Keywords: Speech recognition · Speech synthesis · Voice conversion · Data augmentation

1 Introduction

Automatic Speech Recognition (ASR) systems have become integral to numerous applications, ranging from voice-activated assistants to real-time transcription services. As the demand for these systems grows, so does the need for accurate and versatile ASR models capable of understanding diverse speech patterns and dialects. Traditionally, the development of robust ASR systems relies heavily on extensive datasets of real-world audio, which capture the complexity and variability inherent in human speech. However, acquiring such datasets is often resource-intensive and may not always be feasible, especially for languages or dialects with limited speaker resources.

As an alternative, synthetic data generation has gained traction, leveraging advancements in Text-to-Speech (TTS) technologies and voice conversion

methods to augment training datasets. TTS offers a straightforward approach to synthetic data generation but often falls short of replicating the nuanced attributes of natural speech, resulting in suboptimal ASR performance. In contrast, voice conversion techniques have shown promise in enhancing synthetic speech quality, with models like KNNVC, Seed-VC, and Vec2Wav2 introducing sophisticated transformations aimed at mimicking real-world audio characteristics more closely.

This study evaluates the effectiveness of these diverse data sources—real-world audio, TTS, and advanced voice conversion methods—in training ASR systems. By analyzing the performance impacts of these data types through metrics such as Word Error Rate (WER) and Character Error Rate (CER), we aim to delineate the strengths and limitations of each approach. Our findings highlight the crucial role of authentic data while demonstrating the potential of voice conversion techniques like Seed-VC to substantially improve synthetic alternatives, thus offering a pathway to more accessible ASR solutions in scenarios where real data availability is constrained.

Through this exploration, we seek to inform the broader ASR research community of the practical implications and future directions in synthetic data utilization, inspiring further innovation in the field of speech technology development.

Our study makes several novel contributions to the field of ASR:

Domain-Specific Experimentation: Unlike previous research efforts that predominantly focus on the Common Voice dataset, our experiments are conducted within the specific context of the call center-banking domain. This approach allows for a more targeted analysis and evaluation of ASR models within a real-world, industry-relevant setting that presents unique challenges and opportunities.

Comprehensive Data Analysis: We provide a detailed examination and comparison of real and generated audio data using a speech quality model. This analysis sheds light on the distributional characteristics and quality metrics of each dataset, thereby offering insights into the potential impact on ASR performance and the fidelity of synthetic speech generation.

Comparative Evaluation of Synthesis Techniques: Our work includes a comparative analysis of Text-to-Speech (TTS) and three distinct voice cloning models. This evaluation highlights the differences in speech quality, diversity, and distribution, providing a nuanced understanding of the strengths and limitations of each approach in replicating real-world audio characteristics and improving ASR accuracy.

2 Related Work

Modern Text-to-Speech (TTS) synthesis has been transformed by deep learning, moving from rigid concatenative methods to flexible neural models. The dominant paradigm for a long time was a two-stage pipeline, exemplified by Tacotron

2 [20], which first converts text into a mel-spectrogram, followed by a neural vocoder like WaveNet [16] or HiFi-GAN [10] to generate the final waveform. To address the slow, autoregressive nature of these models, non-autoregressive systems like FastSpeech 2 [19] were introduced, enabling faster inference. A significant leap in quality and efficiency came with end-to-end models like VITS [9], which jointly train the text-to-spectrogram and vocoder components in a single network using variational inference and generative adversarial networks. The most recent frontier treats TTS as a language modeling problem. Models such as VALL-E [21] and Spear-TTS [8] quantize audio into discrete tokens and use a transformer architecture to predict these tokens from text, enabling remarkably human-like synthesis and in-context learning capabilities.

Voice Cloning (VC) aims to generate speech in a specific target voice, typically from a small audio sample. This is pursued through two main paradigms. The first and most common for synthesis from text is based on multi-speaker Text-to-Speech (TTS). This approach, popularized by systems like SV2TTS [7], conditions a TTS model on a speaker embedding extracted from reference audio, enabling zero-shot cloning for speakers unseen during training.

A second, parallel methodology is direct Voice Conversion, which transforms the speaker identity of a source audio utterance into that of a target speaker while preserving the linguistic content. This field includes classic non-parametric methods like kNN-VC [3], which performs conversion by finding the k-nearest neighbors of source speech features within a target speaker’s feature space. More recent deep learning models like AutoVC [18] have achieved high-quality, zero-shot conversion by using an autoencoder with a carefully designed information bottleneck to disentangle speech content from speaker identity. The ultimate goal for both paradigms is high-fidelity speaker mimicry, with the latest generation of TTS models like VALL-E [21] blurring the lines by using reference audio as an acoustic prompt to clone not only timbre but also prosody and acoustic environment with remarkable realism.

The capacity of these systems to generate linguistically identical content across a multitude of cloned speaker identities is particularly valuable for data augmentation, as it can be used to train ASR models that are more robust to speaker variability. Li et al. [11] proposed augmenting ASR training data with synthetic speech generated by a Tacotron 2-based TTS system enhanced with Global Style Tokens (GST), enabling multi-speaker modeling and prosody variation. They synthesized a complete version of the LibriSpeech training set and combined it with natural speech in a 1:1 ratio, training deep convolutional models (Wave2Letter+) with up to 54 layers. This approach led to significant WER improvements, demonstrating that large-scale synthetic augmentation with controlled speaking styles can serve as an effective regularization method and improve generalization in end-to-end ASR models. Baas and Kamper [2] proposed a VC framework for data augmentation aimed at extremely low-resource settings. Their architecture factorizes the speech signal by employing a pretrained CPC-based feature extractor, a quantization-based content encoder to preserve linguistic information, and a hierarchical global style token (HGST)

module to capture speaker identity. They demonstrated the cross-lingual transferability of this approach, showing that a model trained exclusively on high-resource English data could generate synthetic training data for unseen languages, thereby improving the performance of fine-tuned wav2vec 2.0-based ASR systems. Casanova et al. [4] proposed a cross-lingual data augmentation pipeline for ASR leveraging the YourTTS model, a zero-shot multi-speaker TTS and voice conversion system trained with only one speaker in the target language. By synthesizing training data using cross-lingual speaker embeddings and applying audio augmentations, they achieved over 30% absolute WER improvement in Brazilian Portuguese and Russian, demonstrating the feasibility of training ASR models in extremely low-resource conditions. Ogun et al. [15] conducted a comprehensive analysis of synthetic data augmentation for ASR, using flow-based multi-speaker TTS and VC models to isolate the effect of augmenting specific speech attributes such as phonetic content, speaker identity, pitch, and duration. They showed that carefully controlling phoneme diversity, speaker embeddings, and environmental conditions leads to up to 35% relative WER reduction on LibriSpeech, while naïve mixing of real and synthetic data often underperforms due to mismatched distributions.

3 Methodology

3.1 Data

The dataset most frequently utilized for Automatic Speech Recognition (ASR) tasks is the Common Voice (CV) dataset [1], which is publicly accessible and widely employed for ASR data augmentation studies [4, 15]. Nevertheless, there remains a significant gap in the exploration of custom domain data. In particular, this study investigates the audio domain of call centers, specifically for the Turkish language, using internally collected data. This call center data serves as a foundational benchmark for real-world data.

Access to a dataset comprising a diverse range of speakers is crucial for generating speech samples with varied vocal characteristics. For this purpose, we utilized the Librispeech dataset [17], which includes a total of 1281 speakers.

The initial step in creating a training dataset involves generating speech data via off-the-shelf text-to-speech (TTS) technology. We employed Azure-TTS [13] to produce 20 h of speech, utilizing transcriptions from authentic data. The voices “EmelNeural” and “AhmetNeural” were selected at random for this purpose. Hence, the number of speakers is initially 2. This dataset is referred to as “TTS” in the results. Subsequently, these generated speech samples were adapted to different speakers using various voice cloning models. For faster adaptation, we follow Algorithm 1. After cloning, the number of speakers became 1281. Test data does not include synthetic data and contains 5 h of real speech.

3.2 ASR Model

Conformer [5] is a hybrid architecture that integrates convolutional neural networks with transformers, specifically designed to excel in sequence modeling

Algorithm 1. Speaker Selection for TTS Voice Conversion

```

1: Input: Total Speakers  $N$ , Audios per Speaker  $M = 100$ 
2: Output: Converted Audios with Selected Speaker
3: for each unprocessed audio batch do
4:   Randomly select a speaker  $S_i$  from the pool of  $N$  speakers
5:   for each audio  $A_j$  in the batch of  $M$  audio samples do
6:     Convert  $A_j$  to use the voice of selected speaker  $S_i$ 
7:   end for
8:   Log conversion details
9: end for

```

tasks like automatic speech recognition. It combines multi-head self-attention to capture long-range dependencies with convolutional layers for local feature extraction, offering a balance between global context and local detail processing. This combination allows the Conformer to effectively address the limitations of standard transformer models in handling sequential data by enhancing both temporal and spatial feature understanding.

Zipformer [22] is an adaptive architecture optimized for efficiency in computational and memory resources, tailored for real-time tasks such as speech and language processing. It features a dynamic mechanism that adjusts its layer complexity based on the input data, allowing it to manage resources effectively by compressing or expanding layers as needed. This adaptability not only improves processing speed and reduces latency but also makes the Zipformer particularly suitable for deployment in resource-constrained environments, maintaining performance while conserving computational power.

3.3 Voice Conversion Models

K-Nearest Neighbors Voice Conversion (KNN-VC). [3] is a voice conversion framework that utilizes the simplicity and effectiveness of the k-nearest neighbors algorithm to transform speaker characteristics. The model operates by first extracting acoustic features from both the source and target voice samples, typically using methods like Mel-frequency cepstral coefficients (MFCCs). During the conversion process, KNN-VC searches for the k-nearest feature vectors in the target dataset that match the source feature vectors. The transformation is accomplished by averaging these neighbors to estimate the corresponding target features, thereby adapting the source speech to closely resemble the target speaker’s attributes while preserving the original linguistic content. This approach is data-efficient as it does not require the extensive training typical of deep learning models, making it adaptable and easy to implement for real-time applications with moderate computing resources.

Seed-VC. [12] is a pioneering voice conversion system designed to generate high-quality transformations even with minimal training data, known as “seed” data. The model is constructed around advanced neural network architectures,

such as GANs (Generative Adversarial Networks) or VAEs (Variational Autoencoders), which allow it to generalize from small datasets. By leveraging these sophisticated techniques, Seed-VC effectively learns to embody and reproduce the speaker-specific attributes such as pitch, timbre, and prosody of the target voice from limited input data. This is achieved through a fine-tuning process where the network parameters are adjusted to optimize the conversion quality. Seed-VC’s ability to deliver personalized and convincing voice transformations using minimal samples makes it particularly advantageous in scenarios with data scarcity, enabling applications in personalization and customized voice applications across various languages and accents.

Vec2wav2. [6] employs WavLM and vq-wav2vec to encode the source speech. By utilizing these technologies, the speech is encoded for both prompting and content purposes, respectively. Subsequently, the encoded speech is converted into discrete tokens, which are then softened using a Conformer encoder. This process enhances the features’ flexibility and adaptability. Finally, the softened features are resynthesized into high-quality audio through the use of a BigVGAN vocoder, thereby producing output that closely resembles natural speech.

The three voice conversion models exhibit distinct approaches and strengths. KNN-VC utilizes the simplicity of the k-nearest neighbors algorithm, relying on feature matching and averaging, making it data-efficient and suitable for real-time applications without demanding significant computational resources. Seed-VC employs advanced neural architectures like GANs and VAEs, excelling in scenarios with minimal data by learning to emulate speaker-specific characteristics through sophisticated fine-tuning processes. In contrast, vec2wav2 uses technologies such as WavLM, vq-wav2vec, and BigVGAN to encode and resynthesize speech, emphasizing output quality with moderate data requirements. Each model offers unique advantages tailored to specific environments, balancing between data efficiency, adaptability, and high-quality audio output.

The criteria for selecting voice conversion models in this study are multifaceted and carefully considered to ensure optimal performance and applicability. Firstly, the models must be cross-lingual, a feature that enables them to effectively handle and convert speech across different languages. This capability is particularly important for research that involves diverse linguistic datasets, such as Turkish call center audio, which can benefit significantly from cross-lingual agility. Secondly, the models are required to exhibit state-of-the-art accuracy in speech generation. This criterion ensures that the models produce high-fidelity outputs that closely mimic the natural characteristics of human speech, thereby enhancing the realism and effectiveness of the training dataset for the ASR system. Lastly, the voice conversion models need to demonstrate proficiency with short reference speech durations. This capability is crucial for practical applications, as it allows for accurate voice conversion even when limited speech samples are available. This can be particularly advantageous in scenarios where extensive audio data are not readily accessible or where rapid conversion is desired. Together, these criteria make the selected models well-suited for the purposes

of this study, ensuring both the quality and versatility of the generated speech data.

Mesauring Generated Data. We evaluate the similarity between real-world audio data and speech data generated through synthetic means. To accomplish this, we employ the NISQA framework [14], which provides a robust metric for assessing the quality of both authentic and artificially generated speech data. By leveraging NISQA, we can quantitatively examine the distribution of speech quality within both data sets, thereby facilitating a detailed comparison of their respective characteristics. This approach allows us to observe not only the individual quality metrics but also the overall distribution patterns, thereby revealing pertinent differences between real and synthetic speech. Such insights are crucial for understanding the effectiveness and limitations of synthetic data in replicating the nuances of real-world speech.

4 Experiments

Initially, the specified ASR model is trained using the real-world dataset with varying durations of audio data. Following this phase, the training dataset is substituted with speech data generated through TTS and VC models.

Speech Qualities. The comparative distributions of real and generated data qualities are illustrated in Fig. 1. The real-world audio data exhibit greater variability and a higher level of noise compared to the generated speech samples, which display significantly less diversity. The Text-to-Speech (TTS) generated speeches tend to cluster within a high-quality range. However, their distribution does not align with that of the real data, indicating a disparity in variability and naturalness.

Conversely, the speech samples produced through cloning techniques exhibit a distribution that more closely resembles the variability found in real-world data. By converting TTS-generated speech to match the vocal characteristics of speakers from the Librispeech dataset, the resulting data distribution becomes more similar to the authentic data. This observation suggests that voice conversion techniques enhance the representativeness and variability of synthetic speech, thereby bridging the gap between generated and real audio data.

Consequently, this finding supports the hypothesis that voice conversion can be more effective than TTS alone in aiding speech recognition systems to reduce word error rates. The improved alignment of the cloned speech distribution with real data exemplifies its potential to enhance the robustness and accuracy of ASR systems in practical applications.

Experimental Setup for ASR Model. ASR model is inially trained on english data, and then finetuned with the specified cases. Byte Pair Encoding (BPE) is employed for tokenization, utilizing a vocabulary size of 500 tokens.

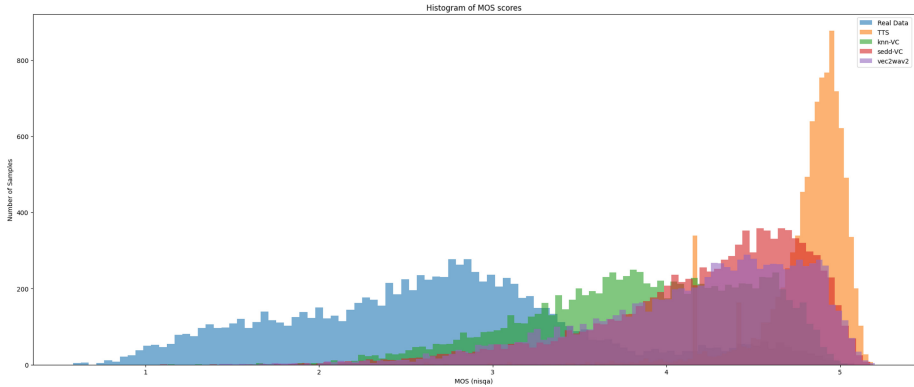


Fig. 1. Histogram of NISQA-predicted MOS (Mean Opinion Score) values for real and synthetic speech data. The “Real Data” (blue) shows a broad distribution of perceived quality, reflecting diverse acoustic conditions. The TTS system (orange) yields consistently high scores with a sharp peak near 5.0, indicating very high synthetic quality but limited variability. Voice conversion methods (knn-VC in green, sedd-VC in red, and vec2wav2 in purple) produce more naturally distributed scores, with sedd-VC and vec2wav2 achieving high MOS values while preserving greater variability than TTS. These results highlight the trade-off between synthetic clarity and realism across systems. (Color figure online)

The context size is configured to 2. For performance evaluation, we conduct an averaging of the final epochs, specifically from the latest 1st, 3rd, 5th, 7th, and 9th epochs, and report the optimal result obtained. We use K2¹ and Icefall² tools for training.

4.1 Real Data and TTS

The results in Table 1 provide a comparison of Word Error Rate (WER) and Character Error Rate (CER) for different datasets used in training an Automatic Speech Recognition (ASR) system over a 20-h period. As expected, the real data outperforms synthetic methods, achieving a WER of 34.40% and a CER of 15.33%. This highlights the effectiveness of real-world audio in capturing the nuances and variability inherent in natural speech, which is vital for developing robust ASR models.

Text-to-Speech (TTS) stands out with a notably high WER of 97.10% and a CER of 80.18%, suggesting significant limitations in replicating the natural qualities of human speech required for accurate recognition. The synthetic speech generated by TTS lacks the variations and natural characteristics, leading to a reduced performance.

¹ <https://github.com/k2-fsa/k2>.

² <https://github.com/k2-fsa/icefall>.

Table 1. Word Error Rate (WER) comparison for different types of training data in an Automatic Speech Recognition (ASR) system, evaluated over training duration of 20 h. The results highlight the effectiveness of real-world audio data and various synthetic speech generation methods, including TTS and voice conversion techniques such as KNNVC, Seed-VC, and Vec2Wav2.

Train Data	CER ↓	WER ↓
Real Data	15.33	34.40
TTS	80.18	97.10
KNNVC	55.08	83.06
Seed-VC	43.97	75.53
Vec2Wav2	46.62	78.42

KNNVC, a voice conversion method, achieves a WER of 83.06% and a CER of 55.08%. This technique surpasses TTS in performance, indicating its ability to generate more natural-sounding synthetic speech, yet it still trails substantially behind real data. Seed-VC yields better results, with a WER of 75.53% and a CER of 43.97%. The relative improvement in WER over TTS is approximately 22.2%, suggesting Seed-VC’s superior capability in synthesizing speech that faithfully mimics the qualities of real-world audio.

Vec2Wav2 demonstrates moderate effectiveness with a WER of 78.42% and a CER of 46.62%, providing improvements over both TTS and KNNVC but slightly underperforming compared to Seed-VC. The relative improvement in WER compared to TTS is approximately 19.2%, highlighting Vec2Wav2’s potential in enhancing synthetic speech for ASR training. Overall, these findings indicate that while real audio data remains crucial for optimal ASR system performance, advanced voice conversion techniques, particularly Seed-VC, can significantly boost the utility of synthetic datasets in speech recognition applications.

4.2 Substituting Real Speech Data with Synthetic Clones via Voice Conversion Models

The presented results in Table 2 provide an insightful examination of Word Error Rate (WER) and Character Error Rate (CER) when utilizing enhanced voice conversion models and their combinations with Text-to-Speech (TTS) in training Automatic Speech Recognition (ASR) systems over a 20-hour duration. Compared to our previous discussion tables, these configurations demonstrate the effectiveness of repeated applications of KNNVC, Seed-VC, and Vec2Wav2 voice conversion methods, both independently and integrated with TTS-generated speech.

Voice conversion techniques alone display varying levels of effectiveness when applied repeatedly. KNNVC applied five times results in the highest WER of 82.25%, indicating a significant error rate that reflects potential overfitting problems or artifacts introduced through excessive synthesis transformations. This

Table 2. Word Error Rate (WER) analysis of speech recognition performance using enhanced voice conversion models and combinations with Text-to-Speech (TTS). Evaluated over training duration of 20 h, this table presents results for models trained on synthetic speech generated through repeated applications of KNNVC, Seed-VC, and Vec2Wav2, as well as the integration of TTS-generated speech with these voice conversion techniques.

Train Data	CER ↓	WER ↓
KNNVC ×5	52.97	82.25
Seed-VC ×5	41.47	73.99
Vec2Wav2 ×5	44.4	75.77
TTS + KNNVC	52.19	81.8
TTS + Seed-VC	43.95	74.62
TTS + Vec2Wav2	46.42	76.82

suggests that without the refinement of TTS integration, pure voice conversion might lack the necessary naturalness or variability found in real-world speech.

Seed-VC applied five times achieves a notable reduction in WER to 73.99%, representing a relative improvement of approximately 9.92% compared to KNNVC. This suggests that Seed-VC’s repeated transformations produce synthetic speech that is closer in acoustic features to real speech, thereby improving ASR model performance.

Vec2Wav2 repeated application, resulting in a WER of 75.77%, offers a more moderate improvement compared to KNNVC, with a relative reduction of around 7.86%. This demonstrates that while Vec2Wav2 improves performance, it yields less pronounced gains than Seed-VC in its fivefold application, highlighting the specific strengths and limitations inherent in each voice conversion model.

The integration of TTS with voice conversion techniques reveals additional enhancements. The combination of TTS with KNNVC leads to a slight reduction in WER to 81.8%, yielding a minor relative improvement of approximately 0.55% over KNNVC alone. However, the benefits are more pronounced when TTS is combined with Seed-VC, where WER reduces further to 74.62%, a relative improvement of about 9.33% compared to TTS with KNNVC, underscoring the complementary nature of Seed-VC and TTS.

Similarly, while combining TTS with Vec2Wav2 results in a WER of 76.82%, this configuration also provides a small yet notable relative improvement of around 5.48% compared to Vec2Wav2 alone. This indicates that TTS can enhance the performance of voice conversion techniques, though the extent varies depending on the particular model.

Overall, the results suggest that voice conversion models, particularly when integrated with TTS, can improve speech recognition accuracy. Nonetheless, the degree of improvement is contingent on the model and approach used, with Seed-VC demonstrating the most substantial gains, especially when paired with TTS.

These findings highlight the critical role of strategic integration and moderation in applying voice conversion and TTS techniques to optimize ASR systems' performance effectively.

4.3 Augmenting Real Speech Data with Synthetic Clones

The results in Table 4 illustrate the effectiveness of different combinations of TTS and voice conversion techniques in reducing the Word Error Rate (WER) in Automatic Speech Recognition (ASR) systems with 20 h of training data. Each configuration demonstrates a considerable improvement over the assumed baseline WER of 34.40% obtained with real data alone.

Integrating TTS with the Vec2Wav2 model yields the most significant reduction, achieving a WER of 28.65%. This translates to a relative improvement of approximately 16.74% compared to the baseline, highlighting the efficacy of this combination in enhancing speech recognition accuracy. However, applying Vec2Wav2 fivefold increases the WER to 30.26%, which still represents a relative improvement of about 12.04% but points to potential over-processing issues.

Similarly, the TTS plus KNNVC configuration results in a WER of 28.98%, corresponding to a relative improvement of approximately 15.78%. Yet, when KNNVC is applied five times, the WER rises to 30.26%, reducing the relative improvement to roughly 12.04%, consistent with findings for multiple Vec2Wav2 applications.

The integration of TTS with Seed-VC results in a WER of 32.30%, yielding a modest improvement of about 6.10%. However, applying Seed-VC fivefold provides a notable WER of 29.71%, improving the relative figure to 13.63%, suggesting that multiple applications can be beneficial for Seed-VC, contrary to trends observed with other techniques.

Finally, using all clones with TTS results in a WER of 29.01%, representing a relative improvement of approximately 15.67%. This balanced approach delivers robust performance without the complexities associated with multiple applications of a single technique.

In summary, these results clearly demonstrate that while all configurations enhance WER over the baseline, the degree of improvement is variable. The Vec2Wav2 technique, notably in its single-application form, produces the most substantial relative gains. Importantly, the analysis suggests that applying voice conversion techniques excessively can lead to less pronounced improvements, potentially due to overfitting or introduction of recognition-inhibiting artifacts. Consequently, selecting the optimal number of applications is crucial for maximizing the benefits of data augmentation in ASR systems (Table 3).

4.4 Exploring the Impact of Varying Training Durations on Model Performance

The analysis of the Word Error Rate (WER) and Character Error Rate (CER) across different training durations (10, 20, and 50 h) highlights several key observations on the performance enhancements achieved through data augmentation

Table 3. Performance comparison of Word Error Rate (WER) when blending real and synthetic training data in Automatic Speech Recognition (ASR) systems. The table highlights WER improvements with real data alone and in combination with TTS and multiple iterations of voice conversion techniques (KNNVC, Seed-VC, Vec2Wav2). Evaluated over 20 h of training data, configurations include single and multiple applications (denoted as $\times 5$) of voice conversion methods.

Train Data	CER ↓	WER ↓
Real Data + TTS + KNNVC	12.32	28.98
Real Data + TTS + KNNVC $\times 5$	13.71	30.26
Real Data + TTS + Seed-VC	13.71	32.30
Real Data + TTS + Seed-VC $\times 5$	13.25	29.71
Real Data + TTS + Vec2Wav2	12.12	28.65
Real Data + TTS + Vec2Wav2 $\times 5$	13.51	30.26
Real Data + TTS + all clones	12.58	29.01

techniques, specifically TTS and various voice conversion models such as KNNVC, Seed-VC, and Vec2Wav2, in the context of Turkish call center data.

Initially examining the WER of models trained solely on real data, a clear and substantial decrease is observed as the training duration increases: from 58.71% with 10 h of training to 34.40% with 20 h, and further to 25.48% with 50 h. This expected improvement underscores the importance of extensive training datasets in enhancing speech recognition models.

When comparing models augmented with TTS and voice conversion techniques to those trained only on real data, significant improvements are observed. With 10 h of training, the integration of TTS and KNNVC, Seed-VC, and Vec2Wav2 results in relative WER reductions of approximately 19% to 21%. Specifically, the WER decreases from 58.71% to about 46–47%, depending on the specific voice conversion model used. At 20 h, the relative improvement in WER is between 6% and 17%, with the augmented models achieving WERs ranging from 28.65% (Vec2Wav2) to 32.30% (Seed-VC), notable reductions from the baseline of 34.40%.

Interestingly, as the amount of training data increases to 50 h, the relative improvements in WER obtained from the voice conversion models become less pronounced, with improvements around 1% to 3% when compared to models trained on real data alone. All augmented models achieve WERs around 25%, closely aligning with the baseline and with one another, indicating diminishing returns on WER improvements from these specific techniques as training duration grows.

These findings reveal that while TTS and voice conversion models initially provide substantial advantages, particularly with limited data, their relative benefit decreases as the training datasets expand. This suggests that the effectiveness of these augmentation techniques is more pronounced in resource-constrained scenarios. Overall, the results underscore the utility of data augmentation in

Table 4. An In-Depth Analysis of Character Error Rate (CER) and Word Error Rate (WER) for Various Voice Conversion Models, Evaluating Performance Across Different Training Durations: A Comprehensive Comparison for 10, 20, and 50 h of Training on Real Data, Augmented with TTS and Different Voice Conversion Techniques Including KNN-VC, Seed-VC, and Vec2Wav2.

Train Data	Amount of Hours in Training					
	10 h		20 h		50 h	
	CER	WER	CER	WER	CER	WER
Real Data	35.57	58.71	15.33	34.40	10.44	25.48
Real Data + TTS + KNNVC	24.38	47.08	12.32	28.98	10.18	25.47
Real Data + TTS + Seed-VC	23.89	46.56	13.71	32.30	10.19	25.19
Real Data + TTS + Vec2Wav2	24.50	46.72	12.12	28.65	10.04	25.26

reducing WER and CER for Turkish call center data, emphasizing their role in enhancing model robustness and accuracy, especially in the early stages of model development.

5 Conclusion

This study investigates the efficacy of various data sources for training Automatic Speech Recognition (ASR) systems, focusing on real-world audio, Text-to-Speech (TTS), and advanced voice conversion techniques such as KNNVC, Seed-VC, and Vec2Wav2. Our findings unequivocally establish that real-world audio data yields superior ASR performance, achieving the lowest Word Error Rate (WER) and Character Error Rate (CER). This underscores the critical role of authentic speech in capturing the nuanced variability essential for robust ASR models.

While synthetic methods offer a feasible alternative in the absence of extensive real data, their effectiveness varies significantly. The TTS approach, though widely accessible, demonstrated substantial limitations, as evidenced by its higher WER and CER, likely due to inadequate representation of natural speech dynamics.

Conversely, the integration of voice conversion techniques presented marked improvements. Seed-VC emerged as the most promising method among the synthetic approaches, delivering the highest reduction in WER and CER relative to TTS, thereby demonstrating its potential to bridge the performance gap between synthetic and real data. Vec2Wav2 also showed efficacy, albeit to a lesser extent. These results suggest that advanced voice conversion can enhance the realism and utility of synthetic datasets for ASR training.

In conclusion, while real-world audio remains the benchmark for optimal ASR performance, the strategic implementation of sophisticated voice conversion methods presents a viable path to enhancing ASR systems, particularly in resource-constrained environments. Future research could further refine these

synthetic methods and explore their broader application in diverse linguistic contexts, paving the way for more accessible and effective ASR technologies.

References

1. Ardila, R., et al.: Common voice: a massively-multilingual speech corpus (2020). <https://arxiv.org/abs/1912.06670>
2. Baas, M., Kamper, H.: Voice conversion can improve asr in very low-resource settings. In: Interspeech (2021). <https://api.semanticscholar.org/CorpusID:242757430>
3. Baas, M., van Niekirk, B., Kamper, H.: Voice conversion with just nearest neighbors (2023). <https://arxiv.org/abs/2305.18975>
4. Casanova, Eet al.: Asr data augmentation in low-resource settings using cross-lingual multi-speaker tts and cross-lingual voice conversion (2023). <https://arxiv.org/abs/2204.00618>
5. Gulati, A., et al.: Conformer: convolution-augmented transformer for speech recognition (2020). <https://arxiv.org/abs/2005.08100>
6. Guo, Y., et al.: vec2wav 2.0: advancing voice conversion via discrete token vocoders (2024). <https://arxiv.org/abs/2409.01995>
7. Jia, Y., et al.: Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Adv. Neural Inf. Process. Syst.* **31** (2018)
8. Kharitonov, E., et al.: Speak, read and prompt: high-fidelity text-to-speech with minimal supervision. *Trans. Assoc. Comput. Linguist.* **11**, 1703–1718 (2023)
9. Kim, J., Kong, J., Son, J.: Conditional variational autoencoder with adversarial learning for end-to-end text-to-speech. In: International Conference on Machine Learning, pp. 5530–5540. PMLR (2021)
10. Kong, J., Kim, J., Bae, J.: Hifi-gan: generative adversarial networks for efficient and high fidelity speech synthesis. *Adv. Neural. Inf. Process. Syst.* **33**, 17022–17033 (2020)
11. Li, J., Gadde, R., Ginsburg, B., Lavrukhin, V.: Training neural speech recognition systems with synthetic speech augmentation. arXiv preprint [arXiv:1811.00707](https://arxiv.org/abs/1811.00707) (2018)
12. Liu, S.: Zero-shot voice conversion with diffusion transformers (2024). <https://arxiv.org/abs/2411.09943>
13. Microsoft: Azure text-to-speech (2023). <https://azure.microsoft.com/services/cognitive-services/text-to-speech/>. Accessed 13 May 2025
14. Mittag, G., Naderi, B., Chehadi, A., Möller, S.: Nisqa: a deep cnn-self-attention model for multidimensional speech quality prediction with crowdsourced datasets. In: Interspeech. ISCA (2021). <https://doi.org/10.21437/interspeech.2021-299>. <http://dx.doi.org/10.21437/Interspeech.2021-299>
15. Ogun, S., Colotte, V., Vincent, E.: An exhaustive evaluation of tts- and vc-based data augmentation for asr (2025). <https://arxiv.org/abs/2503.08954>
16. Oord, A.V.D., et al.: Wavenet: a generative model for raw audio. arXiv preprint [arXiv:1609.03499](https://arxiv.org/abs/1609.03499) (2016)
17. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5206–5210. IEEE (2015)
18. Qian, K., Zhang, Y., Chang, S., Yang, X., Hasegawa-Johnson, M.: Autovc: zero-shot voice style transfer with only autoencoder loss. In: International Conference on Machine Learning, pp. 5210–5219. PMLR (2019)

19. Ren, Y., et al.: FastSpeech 2: fast and high-quality end-to-end text to speech. arXiv preprint [arXiv:2006.04558](https://arxiv.org/abs/2006.04558) (2020)
20. Shen, J., et al.: Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4779–4783. IEEE (2018)
21. Wang, C., et al.: Neural codec language models are zero-shot text to speech synthesizers. arXiv preprint [arXiv:2301.02111](https://arxiv.org/abs/2301.02111) (2023)
22. Yao, Z., et al.: Zipformer: a faster and better encoder for automatic speech recognition (2024). <https://arxiv.org/abs/2310.11230>



Best Data is more Supervised Data – Even for Hungarian ASR

Gergely Dobsinszki^{1,3}, Péter Mihajlik^{2,3(✉)}, Máté Soma Kádár^{1,3},
Tibor Fegyó^{1,2}, and Katalin Mády³

¹ SpeechTex Ltd., Budapest, Hungary
{dobsinszki,kadar}@tmit.bme.hu

² Department of Telecommunications and Artificial Intelligence, Budapest University
of Technology and Economics, Budapest, Hungary

{mihajlik,fegyo}@tmit.bme.hu

³ ELTE Research Centre for Linguistics, Budapest, Hungary
mady@nytud.hu

Abstract. This study aims to improve the accuracy of Hungarian automatic speech recognition (ASR) by applying large amounts of Hungarian training data both for self-supervised learning (SSL) and traditional supervised learning methods. In our experiments, the effectiveness of self-supervised pretraining on both smaller public and larger proprietary datasets was tested. Introducing SSL techniques to small Hungarian training sets resulted in noticeable improvements in model accuracy. When fine-tuning on large datasets containing thousands of hours of Hungarian speech, SSL accelerated training convergence, but fine-tuned models pretrained in English in a supervised way could not be outperformed in terms of word error rate. However, models trained or fine-tuned on a larger-than-ever purely Hungarian dataset achieved state-of-the-art accuracy across multiple independent evaluation sets.

Keywords: Deep learning · ASR · Self-supervised learning · Acoustic modeling

1 Introduction

Previous research has shown that starting neural network training from pre-trained weights rather than random initialization can improve convergence speed [3]. This is also true for automatic speech recognition (ASR), where even weights from a model trained on a different language can provide a good initialization point [11].

The increasing size of neural models demands ever-larger amounts of data, which cannot be met through costly and slow manual annotations. Self-supervised learning (SSL) offers a solution to this problem by producing initial models/weights capable of high-level representation without output labels. These can then be fine-tuned for specific tasks (e.g., speech-to-text) using a relatively shallow added layer, shortening training and improving final results [16,25].

Our goal was to explore to what extent self-supervised pretraining on Hungarian data can benefit Hungarian ASR. We trained SSL acoustic models on nearly 20,000 h of Hungarian audio, fine-tuned them on various labeled corpora, and evaluated them on public datasets. Compared to other publicly available Hungarian models, our results show broader applicability and, in some cases, significantly higher recognition accuracy.

2 Data

As a first step, it was necessary to collect audio data for SSL training. Compared to supervised training, this method requires orders of magnitude more data. Due to the large volume of data, the application of audio file compression methods became an important consideration. To facilitate data handling, we created tarred collections from individual files. Since these collections bundle multiple files together, they require fewer file read operations, resulting in a more efficient training process.

2.1 SSL

For self-supervised learning, we needed a large amount of Hungarian speech data. While obtaining such data in itself was not a challenge, we prioritized publicly available databases to support reproducibility and comparability of the experiments.

The majority of the training data came from the VoxPopuli V2 dataset [24], maintained by Meta/Facebook Research. This dataset includes European Parliament speeches in various languages from multiple nations. It contains longer recordings (several minutes in length), but since ASR systems require shorter speech segments for training, we split the audio into segments no longer than 20 s. We mostly applied segmentation at automatically detected silent parts that did not contain speech. We also removed longer pauses to reduce the overall length of the dataset. The resulting corpus contains 17,470 h of Hungarian-language recordings.

We also used a relatively smaller in-house (IH) dataset for self-supervised training. The IH corpus mainly consists of radio talk shows. Its preprocessing followed the same procedure as for the VoxPopuli dataset, with silence-based segmentation and trimming, resulting in a total of 3.36 thousand hours of material. Although it is significantly smaller than the Facebook Research corpus in volume, it represents a different speech style and domain, thus contributing valuable diversity to the SSL training process, which was carried out on the combined VoxPopuli + IH dataset.

2.2 Mozilla Common Voice (CV-16)

For fine-tuning and evaluation experiments, we used the Hungarian subset of the freely available Mozilla Common Voice dataset, version 16.1 (CV-16) [1].

This version contains a total of 92 h of recorded, verified data. The durations of the training, validation, and test subsets are 52.5, 16.8, and 17.7 h, respectively.

2.3 BEA

For further fine-tuning experiments, we sought a dataset that includes conversational speech. The BEA (Hungarian Spoken Language Database) [5,20] is a corpus collected and maintained by the Hungarian Research Centre for Linguistics. It contains both spontaneous (monologic and dialogic) and read speech from multiple individual speakers. Within BEA, we used the BEA-Base subset [19], which was specifically created for the evaluation of speech recognition models. The dataset is freely available for research purposes.

We used the official split of BEA-Base, the subset for training (train-114) is 68 h long, the validation set (dev-spont) is 3.8 h, and the test set (eval-spont) is 4.75 h long.

2.4 BNC

The dataset we refer to as BNC (Broadcast News and Conversations) is a proprietary, manually transcribed/verified collection consisting of several thousand hours of audio tracks from Hungarian television broadcasts. Most of the recordings contain multi-speaker conversations, while a smaller portion includes read news or prepared speeches. Prior to use, the audio data was segmented and normalized to ensure consistency and comparability with other training and evaluation sets.

For the BNC dataset, we created a training set of 2,703 h, a validation set of 113 h, and a test set of 41 h of audio.

3 Experiments

The focus of our experiments was on how to best utilize large-scale speech data for training ASR systems. Working with large datasets is naturally cumbersome, so we optimized hyperparameters on the *development* sets of the smaller CV-16 and BEA datasets. In these cases, we deliberately avoided using the designated *evaluation* sets, reserving them for evaluating the training performed on the large (BNC) dataset.

3.1 Experimental Environment

All experiments were conducted using version 1.23 of the NVIDIA NeMo Toolkit [9]. The environment supports various model architectures and enables multi-GPU and multi-node training. SSL training can also be configured within it.

The NeMo Toolkit integrates with the SLURM job scheduler, which was used to run SSL training on the Komondor supercomputer [6].

3.2 Models

When choosing the neural architecture, we ensured that at least one pretrained English-language model was available. Starting from these initial weights, we were able to perform baseline training for comparison with our SSL-based experiments. In self-supervised training, we first created a general model from unlabeled Hungarian data. This model was then fine-tuned for the speech recognition task using various annotated datasets.

We sought a specific acoustic model architecture for our research. Since SSL training is computationally intensive, we aimed to select a model with a relatively low parameter count that could be trained efficiently, while still achieving high accuracy and meeting potential industrial requirements (limited GPU memory, compute capacity, etc.). We therefore chose the *FastConformer Large (CTC)*¹ model, which is conveniently supported in the NeMo Toolkit [9] and has approximately 121 million parameters [22].

The Conformer architecture [8] is essentially a transformer encoder [23] that includes, in addition to the usual attention and feed-forward layers, a 1D convolutional layer in each block to better capture local context. This architecture is arguably the most popular in end-to-end deep neural ASR systems. The FastConformer differs from the classic Conformer in that it replaces the standard 1D convolution with a ‘time-channel separable’ convolution [13], which is a parameter- and computation-efficient approximation of the former.

The output of the FastConformer model is a probability distribution over subword units (with a scalable vocabulary size) every 80 ms, followed by a final CTC [7] layer and loss function. For tokenizing the transcriptions used during fine-tuning, we applied the SentencePiece algorithm [14] with the default vocabulary size of 1024.

3.3 SSL Training

For SSL training, we experimented with both Contrastive [2] and Masked Language Modeling (MLM) [10] loss functions. In contrastive approaches, models learn to distinguish true latent representations from distractors, while in MLM (similar to the training of BERT models [4]), the model must predict a masked audio segment based on the known (unmasked) context.

During SSL training, we used the default configuration of the given NeMo version, and we only list those parameters below that diverged from the default.

For SSL training, we used 2 nodes with a total of 16 A100 SXM4 GPUs, each with 40 GB VRAM. Each node had 64 CPU cores and 256 GB RAM available. One epoch took approximately 1 h. In all our trainings, we used the AdamW optimizer [12, 17].

¹ https://github.com/NVIDIA/NeMo/blob/main/examples/asr/conf/fastconformer/fast-conformer_ctc_bpe.yaml.

4 Results

In the following, we present the results of the training runs that started from SSL-pretrained weights (as described in Sect. 2), alongside results from “cross-language transfer learning” using English-language acoustic models pretrained by NVIDIA as baselines. The Hungarian SSL models and the English (supervisedly pretrained) models were always fine-tuned on Hungarian-language corpora, using the validation set to monitor training. Final evaluations were conducted on test sets independent of the training and validation data.

4.1 SSL + CV-16

In this setting, we experimented with several SSL-pretrained models. Apart from the initial weights, all training parameters were fixed and aligned with the NeMo FastConformer-Large training recipe. The models were trained (fine-tuned) on the CV-16 training set for 100 epochs, with a learning rate of $2 * 10^{-4}$ and a tokenizer size of 256. The results are summarized in Table 1.

Table 1. Fine-tuning English and various SSL-pretrained models on CV-16.

Experiment Name	SSL Loss Type	SSL Epoch	Num Negatives	Val WER
en_weights	NA	NA	NA	16.49 %
ssl_weights_1	Contrastive	70	40	15.56 %
ssl_weights_2	Contrastive & MLM	100	40	13.20 %
ssl_weights_3	Contrastive & MLM	100	100	13.43 %

4.2 SSL + BEA

In this series of experiments, we worked with only one SSL-pretrained model. Our main focus was on tuning the learning rate. As Table 2 shows, shaping the learning rate schedule had a significant impact on the validation WER. Again, a notable improvement (14.53% vs 17.11%, 15% relative) due to SSL pre-training can be observed.

The model labeled ‘ssl_weights_3’ in Table 2 refers to the identically named model in Table 1. This pretrained model was chosen because it showed consistently lower validation WER during fine-tuning on CV-16 (only final results are shown in the table), outperforming the other variants.

4.3 SSL + BNC

As the third and most important experiment, we fine-tuned the selected pre-trained model on the large BNC corpus of transcribed data.

Table 2. Fine-tuning different SSL models on BEA and evaluation on BEA dev_spont.

Initial Weights	Initial LR	Minimum LR	Warmup Steps	Val WER
en_weights	9e-5	5e-8	2000	26.73 %
en_weights	5e-4	5e-8	12500	17.11 %
ssl_weights_3	5e-4	5e-6	1000	17.47 %
ssl_weights_3	4e-4	5e-8	2000	16.46 %
ssl_weights_3	2e-4	5e-8	2000	15.32 %
ssl_weights_3	9e-5	5e-8	2000	14.53 %
ssl_weights_3	8e-5	5e-8	2000	14.76 %

Fine-tuning on BNC was carried out using both the Hungarian SSL-pretrained model (trained on 20,000 h of Hungarian audio) and the supervised English model pretrained by NVIDIA. While previous results (see Tables 1 and 2) clearly favored the Hungarian SSL initialization over the English one, this time we observed the opposite.

Although the SSL-pretrained model converged faster, it did not outperform the supervised English model in final accuracy. Figure 1 shows the validation WER curves for both SSL-based and English-based training, where the SSL-initialized model converges more quickly but ultimately achieves slightly worse test set performance, as confirmed in Table 3. One possible explanation is the lack of BNC-specific hyperparameter optimization.

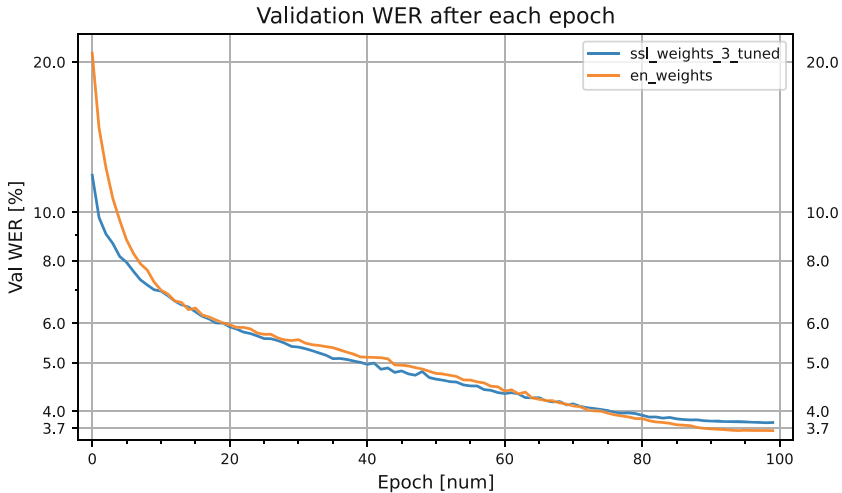


Fig. 1. Training on the BNC dataset starting from English (en_weights) and SSL-pretrained (ssl_weights_3_tuned) weights. The SSL model converges faster but ultimately underperforms the supervised English model.

Table 3. Validation and test results for fine-tuning on the BNC corpus using SSL-pretrained vs. English-supervised weights.

Initial Weights	Val WER (%)	Test WER (%)
en_weights	3.66	3.83
ssl_weights_3	3.75	3.96

We compared the models fine-tuned on BNC to other Hungarian-language neural ASR systems. One such system is BEAST2 [18], trained on a limited amount of in-domain data (BEA) exploiting large scale SSL pre-trained models [2], achieving the best performance on the BEA test set. Another is Whisper [21] (large-v3), a publicly available model by OpenAI capable of multilingual transcription, language detection, and even translation.

As seen in Table 4, both of our models significantly outperform the other two, despite not using a language model (LM), meaning they were trained without any prior knowledge of written Hungarian. Additionally, our architectures have the lowest parameter count among all four systems. The only case where BEAST2 performs better is on the BEA dataset—which is understandable, as BEAST2 was trained on BEA, while our models were not and therefore treat it as out-of-domain. Even so, our models outperform Whisper large-v3 on this dataset, for which BEA is also unseen.

Table 4. Results of BEAST2 [15], Whisper large-v3 [21], and our models (from Hungarian SSL and English supervised pretraining) on three publicly or research-accessible evaluation datasets.

Model	LM	CV-16 WER (%)	FLEURS WER (%)	BEA WER (%)
BEAST2	Yes	19.53	25.63	10.98
Whisper large-v3	Implicit	13.4	12.9	21.70
FastConformer HU-SSL + BNC (own)	No	7.43	9.98	13.06
FastConformer EN + BNC (own)	No	7.32	9.74	12.70

4.4 SL + BNC

As a control, we also trained a few variants of FastConformer HU using classical supervised learning (SL), analogous to the last row of Table 4. Besides the NeMo-recipe Large model, we trained an XL (600M parameters) version using encoder + CTC as well as RNN-Transducer (T) style training, where the FastConformer L and XL encoders were paired with a recurrent decoder. Results are shown in Table 5.

These XL and Transducer models were trained for 50 epochs (as opposed to 100 in earlier experiments), starting from the English-supervised models released by NVIDIA. The first row of Table 5 matches the last row of Table 4. Except for the basic CTC model, all architectures outperform the BEAST2 [15] model on the spontaneous BEA (even though BEA data was not used for training). Moreover, to the best of our knowledge, the final FastConformer-T XL model outperforms all previous state-of-the-art Hungarian ASR results on CV, FLEURS and BEA with a large margin.

Table 5. Evaluation of NVIDIA NeMo En models fine-tuned on the the large Hungarian BNC dataset (2700 h of manually transcribed audio).

Model Architecture	CV-16 WER (%)	FLEURS WER (%)	BEA WER (%)
FastConformer CTC	7.32	9.74	12.70
FastConformer CTC XL	6.54	7.90	10.55
FastConformer-T	7.02	8.89	10.88
FastConformer-T XL	5.45	7.02	9.92

5 Summary

Our results show that when fine-tuning on small datasets, SSL pretraining can bring significant improvements over cross-lingual transfer from pretrained weights in a different language. However, this advantage disappears when fine-tuning is performed on large amounts of transcribed data, at least with the model architecture we examined.

For the self-supervised training, we used less than 20,000 h of audio, which is relatively limited for such approaches. In the future, we plan to expand this dataset and fine-tune such a model in a supervised manner.

We also aim to explore self-supervised training with larger models. These require more VRAM and longer training time, and thus could not be included in the current study. However, prior research has shown that the benefits of SSL pretraining become more prominent with larger, higher-capacity models. Overall, our findings suggest that the best performance can be achieved using large English-supervised pretrained models that are further fine-tuned on large corpora under supervision. Our conclusion is that collecting and applying supervised data remains the most effective method for improving speech recognition accuracy—based on our Hungarian-language experiments.

Acknowledgments. This work was partially supported by the National Research, Development and Innovation Fund of Hungary under projects NKFIH K143075 and K135038, as well as project NKFIH-828-2/2021 (MILAB). We would also like to thank the Governmental Agency for IT Development (KIFÜ²(<https://ror.org/01s0v4q65>))

for providing access to Hungary’s Komondor supercomputer, and NVIDIA for the Academic Hardware Grant.

Disclosure of Interest. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ardila, R., et al.: Common voice: a massively-multilingual speech corpus. arXiv preprint [arXiv:1912.06670](https://arxiv.org/abs/1912.06670) (2019)
2. Baevski, A., Zhou, H., Mohamed, A., Auli, M.: Wav2vec 2.0: a framework for self-supervised learning of speech representations (2020). <https://arxiv.org/abs/2006.11477>
3. Cho, E., Li, J., Kim, S., Jinyu, L.: Cross-language transfer learning and domain adaptation for end-to-end automatic speech recognition. In: Proceedings of the 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7494–7498. IEEE (2020). <https://doi.org/10.1109/ICASSP40776.2020.9053538>
4. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding (2019). <https://arxiv.org/abs/1810.04805>
5. Gósy, M.: Bea-a multifunctional Hungarian spoken language database. *Phonetica* **105**, 50–61 (2013)
6. Governmental Agency for IT Development (KIFÜ): Komondor supercomputer (2023). <https://ncc.dkf.hu/en.html>. Hungary’s most powerful supercomputer, located at the University of Debrecen
7. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
8. Gulati, A., et al.: Conformer: convolution-augmented transformer for speech recognition. arXiv preprint [arXiv:2005.08100](https://arxiv.org/abs/2005.08100) (2020)
9. Harper, E., et al.: Nemo: a toolkit for conversational ai and large language models (2023). <https://nvidia.github.io/NeMo/>
10. Hsu, W.N., Bolte, B., Tsai, Y.H.H., Lakhota, K., Salakhutdinov, R., Mohamed, A.: Hubert: self-supervised speech representation learning by masked prediction of hidden units (2021). <https://arxiv.org/abs/2106.07447>
11. Huang, J., et al.: Cross-language transfer learning, continuous learning, and domain adaptation for end-to-end automatic speech recognition (2020). <https://arxiv.org/abs/2005.04290>
12. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. Kriznan, S., et al.: Quartznet: deep automatic speech recognition with 1d time-channel separable convolutions. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6124–6128. IEEE (2020)
14. Kudo, T., Richardson, J.: Sentencepiece: a simple and language independent subword tokenizer and detokenizer for neural text processing. arXiv preprint [arXiv:1808.06226](https://arxiv.org/abs/1808.06226) (2018)

15. Kádár, M.S., Dobsinszki, G., Mády, K., Mihajlik, P.: Feeding the beast – the latest developments on the BEA Speech Transcriber and its integration with language model – in Hungarian. In: XIX. Hungarian Conference on Computational Linguistics, pp. 135–143 (2023)
16. Lee, Y., Willette, J.R., Kim, J., Hwang, S.J.: Visualizing the loss landscape of self-supervised vision transformer (2024). <https://arxiv.org/abs/2405.18042>
17. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization (2019). <https://arxiv.org/abs/1711.05101>
18. Mihajlik, P., et al.: What kind of multi- or cross-lingual pre-training is the most effective for a spontaneous, less-resourced ASR task? In: Proceedings of the 2nd Annual Meeting of the ELRA/ISCA SIG on Under-resourced Languages (SIGUL 2023), pp. 58–62 (2023). <https://doi.org/10.21437/SIGUL.2023-13>
19. Mihajlik, P., Balog, A., Graczi, T.E., Kohari, A., Tarján, B., Mady, K.: BEA-base: a benchmark for ASR of spontaneous Hungarian. In: Calzolari, N., et al. (eds.) Proceedings of the Thirteenth Language Resources and Evaluation Conference, pp. 1970–1977. European Language Resources Association, Marseille (2022). <https://aclanthology.org/2022.lrec-1.211/>
20. Neuberger, T., Gyarmathy, D., Grácsi, T.E., Horváth, V., Gósy, M., Beke, A.: Development of a large spontaneous speech database of agglutinative hungarian language. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2014. LNCS (LNAI), vol. 8655, pp. 424–431. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10816-2_51
21. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision (2022). <https://arxiv.org/abs/2212.04356>
22. Rekish, D., et al.: Fast conformer with linearly scalable attention for efficient speech recognition (2023). <https://arxiv.org/abs/2305.05084>
23. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
24. Wang, C., et al.: VoxPopuli: a large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, vol. 1: Long Papers, pp. 993–1003. Association for Computational Linguistics, Online (2021). <https://doi.org/10.18653/v1/2021.acl-long.80>. <https://aclanthology.org/2021.acl-long.80>
25. Yang, H., Zhao, J., Haffari, G., Shareghi, E.: Self-supervised rewiring of pre-trained speech encoders: towards faster fine-tuning with less labels in speech processing. In: Goldberg, Y., Kozareva, Z., Zhang, Y. (eds.) Findings of the Association for Computational Linguistics: EMNLP 2022, pp. 1952–1959. Association for Computational Linguistics, Abu Dhabi (2022). <https://doi.org/10.18653/v1/2022.findings-emnlp.141>. <https://aclanthology.org/2022.findings-emnlp.141>



Arabic ASR on the SADA Large-Scale Arabic Speech Corpus with Transformer-Based Models

Branislav Gerazov¹(✉) , Marcello Politi² , and Sébastien Bratières²

¹ FEEIT, Ss Cyril and Methodius University, Rugjer Boshkovikj 18,
1000 Skopje, Macedonia

gerazov@feit.ukim.edu.mk

² Pi School, Via Indonesia 23, 00144 Rome, Italy
{sebastien, marcello.politi}@picampus-school.com

Abstract. Automatic speech recognition (ASR) has seen significant improvements with the advent of deep learning and end-to-end models based on Transformer architectures. Arabic ASR has remained a challenging task due to the language’s complexity, especially in terms of its dialectal variety. We explore the performance of several state-of-the-art ASR models on a large-scale Arabic speech dataset – the SADA (Saudi Audio Dataset for Arabic) comprising 668 h of high-quality audio. The dataset includes multiple dialects and environments, and specifically a noisy subset; all of these make it particularly challenging for ASR. We evaluate the performance of the models on the SADA test set, and we explore the impact of finetuning, language models, as well as noise and denoising on their performance. We find that the best performing model is the MMS 1B model finetuned on SADA with a 4-gram language model that achieves a WER of 40.9% and a CER of 17.6% on the SADA test clean set. We find that the best path towards improving the performance of the models in noise is finetuning them on the noisy data, with denoising adversely impacting performance overall and only leading to improvement the noisiest samples.

Keywords: ASR · Arabic · Transformer · SADA · Wav2Vec · XLSR · Whisper · MMS

1 Introduction

Automatic Speech Recognition (ASR) is a technology that converts spoken language into text. ASR systems have been around for decades, but they have seen a significant improvement in the last few years with the advent of deep learning and the creation of large-scale datasets. The most notable improvements have been achieved with the introduction of end-to-end models, such as Wav2Vec2 [5] and Whisper [17], which are based on Transformer architectures [19].

In this work, we focus on Arabic ASR, which is a challenging task due to the complexity of the Arabic language, especially in terms of its dialectal variety.

Arabic is spoken by more than 400 million people worldwide, and it has many dialects that differ significantly from each other, as well as the Modern Standard Arabic (MSA) language norm. It is also a language that has not been the main-stream focus of ASR research, with limited resources available for training and evaluating ASR systems.

In recent years, there has been growing interest in Arabic ASR, with several datasets and models being developed. Dhouib et al. [9] provide a systematic literature review of Arabic ASR, covering the period from 2011 to 2021 and focusing on the toolkits, datasets, and techniques used in Arabic ASR. They found that KALDI [15] and HTK [21] were the most popular toolkits, and that 89.5% of the studies focused on Modern Standard Arabic (MSA), while 26.3% focused on various Arabic dialects. Abdelhamid et al. [1] in addition provide a review of the development of end-to-end Arabic ASR models. Recently, Besdouri et al. [6] provided a comprehensive overview of Arabic ASR, focusing on the challenges posed by the language’s diverse forms and dialectal variations. To address the challenges of dialectal code-switching in Arabic ASR, Chowdhury et al. [7] propose a multilingual end-to-end ASR system based on the conformer architecture [10] that outperforms state-of-the-art monolingual dialectal Arabic and code-switching Arabic ASR systems. Finally, the Open Universal Arabic ASR Leaderboard has been introduced to benchmark open-source general Arabic ASR models across various multi-dialect datasets [20].

We explore the performance of several state-of-the-art ASR models on a large-scale Arabic speech dataset, the SADA (Saudi Audio Dataset for Arabic) [2], which contains 668 h of high-quality audio from Saudi television shows. The dataset includes multiple dialects and environments, specifically a noisy subset that makes it particularly challenging for ASR. We evaluate the performance of the models on the SADA test set, and we explore the impact of finetuning, language models, as well as noise and denoising on the performance of these models.

2 Dataset

There are several datasets available for Arabic ASR. The Arabic Speech Corpus – a single speaker dataset created for TTS in the Syrian Damask dialect, with some 2k samples [11]. The Egyptian-ASR-MGB-3 dataset of 16 h manually transcribed multi-genre data of Egyptian collected from different YouTube channels. The Tarteel Recitation Dataset of Quranic recitation with 67.4 h from 1,200 speakers [14]. Finally, Mozilla Common Voice [3] that as of v22 contains 157 h of Arabic (92 h validated) with 1,632 speakers.¹

In our work, we focus on the SADA (Saudi Audio Dataset for Arabic) dataset – a large-scale Arabic speech dataset with 668 h of high-quality audio [2]. The audio is sourced from 57 Saudi Broadcasting Authority television shows, suitable for speech recognition training. It includes both read and spontaneous speech in multiple dialects – primarily the three major Saudi dialects (Najdi, Hijazi and Khaleeji), but also including Yemeni, Egyptian, and Levantine. The dataset was

¹ <https://commonvoice.mozilla.org/en/datasets>.

transcribed and prepared by the National Center for Artificial Intelligence in Saudi Arabia.

Our initial data exploration shows SADA to be a challenging dataset. There are quite long samples (>30 s), as well as issues with speaker overlap, and transcription errors. Moreover, less than a third of the data is considered clean, with another third classified as noisy and the final one containing music, as shown from the data spread in Table 1. We see this as a good opportunity to test the performance of state-of-the-art ASR models on a challenging large-scale Arabic dataset, and to explore the impact of finetuning and denoising on the performance of these models.

Table 1. SADA data spread across the environments.

Environment	Train (h)	Valid (h)	Test (h)	Total (h)
Clean	116.19	2.77	3.73	122.69
Music	159.29	4.13	3.17	166.59
Noisy	141.73	2.26	3.84	147.83
Car	0.42	0	0	0.42
Total	417.63	9.16	10.74	437.53

To explore the impact of noise on the performance of the ASR systems, we analyze more closely the contents of the noisy part of the SADA dataset. It appears that most of the data is not that noisy, i.e. sometimes there is some laughter or English words present. Otherwise, there are several type of noise:

- audience noise (including chatter, laughter and applause),
- traffic noise,
- white/brown noise (sometimes low frequency),
- nature sounds, e.g. birds and dogs,
- harmonic noise, i.e. like slow music,
- audio effects, e.g. doors, thumping, objects hitting,
- microphone noise, i.e. microphone handling noise.

3 Methodology

We consider several state-of-the-art ASR models to build our ASR system. We evaluate their performance on SADA w.r.t. WER (Word Error Rate) and CER (Character Error Rate). We then explore improving them via finetuning, adding language models and denoising.

3.1 ASR Models

Our initial pick is the XLSR-53 (Cross-Lingual Speech Representations) model [8] finetuned for Arabic by Mohamed El-Geish as a baseline model for our experiments, which we will refer to as xlsr-elgeish.² The base XLSR-53 model is a

² <https://huggingface.co/elgeish/wav2vec2-large-xlsr-53-arabic>.

multilingual 300 M parameter Wav2Vec2 [5] model pretrained on 50 kh from 53 languages, no Arabic. The elgeish-xlsr model was finetuned on the Arabic Speech Corpus and Common Voice v6.1, which contains 78 h of Arabic (50 h validated) of 672 speakers. The model works with 16 kHz sampled audio and includes no language model (LM).

Next, we consider XLS-R [4] – a Wav2Vec2 model pretrained on 436 kh in 128 languages, including 95 h of Arabic. The XLS-R model comes in three model sizes of 300 M, 1 B and 2 B parameters. We choose the 300 M parameter model for our experiments, which is the same size as elgeish-xlsr.

Finally, we consider the Whisper and MMS (Massively Multilingual Speech) models. Whisper [17] is a family of large transformer-based models pretrained on 680 kh of audio in 97 languages, including 739 h of Arabic. The models come in five sizes, from 39 M (tiny) to 1.55 B (large) parameters. The large model has been updated to v2 that features $2.5\times$ more training epochs and added regularization, and v3 that is trained on a larger dataset of 5+ Mh of audio. We did not use the large model as we chose to limit the analysis to models that can run on a single consumer grade GPU with 8 GB of VRAM. Whisper’s built-in neural sequence decoder acts in part like a language model.

MMS [16] is a 300 M and 1 B parameter Wav2Vec2 model pretrained on 1,406 languages with 491 kh of speech, and then finetuned for ASR in 1,107 languages using 44.7 kh of labeled speech data, and additional adapter layers (2 M parameters) that are finetuned for each language. MMS + LM outperforms Whisper medium and large v2, halving the WER and CER whilst being trained on more than $10\times$ less labeled data and supporting $10\times$ more languages.

3.2 Finetuning

The largest improvements in ASR performance are achieved by finetuning the model on the target dataset. We explore finetuning of the XLSR, XLS-R and MMS models on SADA. In order to maintain a fair comparison, we finetune all models for 100k steps, which is the number of steps used by Elgeish to finetune the XLSR model on the Common Voice v6.1 Arabic dataset.³

For finetuning the XLSR and MMS models we experiment with unfreezing the encoder weights. We also explore finetuning the original XLSR model and not the already finetuned El-Geish XLSR model.

3.3 Language Models

Language models can improve the performance of ASR systems, especially in the absence of finetuning. For example, adding a 4-gram language model to an English Wav2Vec2 ASR model, when the model has only been finetuned with 10 min of speech, decreases the WER on the Librispeech test clean set from 40.2% to 6.6% and adding a larger Transformer-based language model decreases

³ Note that Elgeish first finetuned the XLSR model on the Arabic Speech Corpus, but the number of steps is not disclosed.

it further to 4.8% [5]. In the case of finetuning, the language model leads only to a marginal improvement, i.e. from 2.2% to 2.0% WER and then to 1.8% WER with a larger Transformer-based language model.

In our experiments we choose a 4-g language model in order to avoid increasing the overall ASR system complexity, while still reaping the benefits of a language model. We choose KenLM – an n-gram Language Model with Kneser-Ney smoothing, fast and low-memory querying [12], as well as fast and scalable estimation based on its streaming algorithms [13].

3.4 Noise and Denoising

Since one third of the SADA data is noisy, we explore the impact of noise on the performance of the ASR systems in more detail. We try finetuning the XLSR model on the noisy SADA data, and then evaluate its performance.

We also explore denoising and its potential to improve the performance of the ASR systems. We focus on denoising using spectral gating, as implemented in the Noisereduce algorithm [18] that first computes a spectrogram of a signal and estimates a noise threshold (or gate) for each frequency band of that signal/noise. The threshold is then used to compute a mask, which gates noise below the frequency-varying threshold.

In its non-stationary version, the estimated noise threshold is continuously updated over time. This relies on the fact that most types of noise occur at timescales larger than the timescale of the speech signal. In practice, the spectrogram of the signal is time-smoothed and it is used to calculate a noise mask that is smoothed with a filter over frequency and time.

We use Noisereduce to denoise the SADA noisy data, and evaluate the XLSR model on the denoised data. We also try finetuning the XLSR model on the denoised SADA data, and evaluate its performance on the denoised data.

Finally, we evaluate if denoising helps performance w.r.t. to the noise level. Our hypothesis is that denoising helps more on noisier data, and hurts performance on less noisy data. SADA is a good dataset for this analysis as its noisy set contains samples that span the range from being very noisy and not noisy at all. We use two proxies for the noise in the data:

- the CER, which we assume is inversely correlated with the noise level, and
- the average relative noise spectrogram energy extracted with Noisereduce.

For both of these proxies we calculate the absolute improvement of CER due to denoising as the difference between the CER on the noisy and the CER on the denoised data.

4 Results

The results from using the chosen ASR models on the SADA test clean dataset are reported in Table 2. The table also shows the performance of the models finetuned on the SADA training set, as well as the performance of the models finetuned on the SADA training set with an unfrozen encoder.

Table 2. ASR model performance on SADA clean test set.

Model		Size	WER	CER
Base models	xlsr-elveish	300 M	93.75	53.91
	whisper small	300 M	254.86	215.27
	whisper medium	770 M	116.69	154.70
	mms	1 B	84.40	44.20
+ Finetuning	xlsr-elveish +SADA	300 M	54.34	23.86
	xlsr-elveish +SADA (unfreeze)	300 M	51.64	22.13
	xlsr +SADA	300 M	91.47	51.38
	xls-r +SADA	300 M	90.65	49.22
	mms +SADA (unfreeze)	1 B	51.49	19.90
+ Language models	xlsr-elveish +SADA (unfreeze) +LM	300 M	42.62	18.31
	mms +SADA (unfreeze) +LM	1 B	40.86	17.60

4.1 Base Models

The results obtained when using the ASR models directly without finetuning confirm the intuition we gathered from the data exploration stage that SADA is a challenging dataset. The El-Geish XLSR model gave a WER of almost 100% and a CER of around 54%!

The two Whisper models – small and medium, perform the worst with a WER of 254.9% and 116.7%, and a CER of 215.3% and 154.7%, respectively. On closer inspection, we found on some samples the WER to go up to 15000%! We identified the root cause of these results to be the proclivity of the Whisper model to hallucinate and generate unbound text. The medium size model tends to do less hallucination, explaining the better performance.⁴ An example of this behavior can be seen in the following sample:

[illegible]

⁴ One could expect that the large Whisper model would do less hallucination, but not necessarily so.

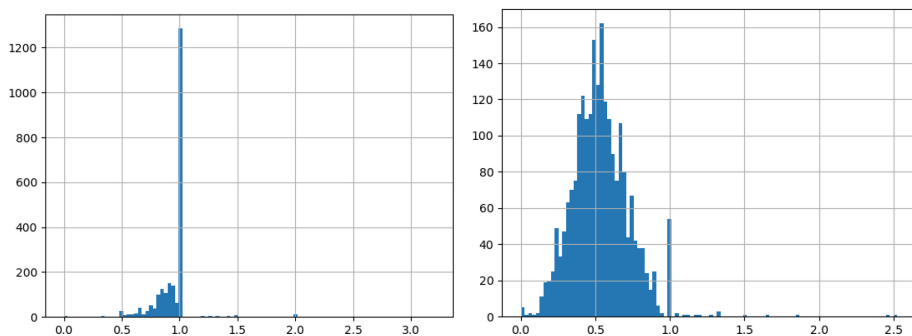


Fig. 1. WER (left) and CER histograms (right) with count on the y -axis for the El-Geish XLSR model on the SADA test clean dataset.

Here, the model not only hallucinates sentences that are not present in the audio, but it also starts repeating them 15 times and then again 8 times!⁵

The MMS 1B model is the largest model we tested. It achieves the best performance with a WER of 84.4% and a CER of 44.2%, which is still quite high.

To get a clearer picture of the error spread across the test clean dataset, we plot a histogram of the WER and CER for the XLSR model in Fig. 1. We can clearly see that the mode for the WER distribution is 100% and it dominates the distribution, with only a small tail going down to 50%. The mode for the CER distribution is around 50%, but the distribution is more spread out and looking like a Gaussian.

4.2 Finetuning

We finetune the ASR models on a subset of the SADA clean training set, which contains samples with lengths between 2s and 10s. We set the lower bound based on the distribution of the CER for short samples, which falls mostly below 100% CER for samples longer than 2s. The upper bound is set to 10s, to allow us to finetune the model on a 8 GB GPU. The finetuning subset contains 35k of the 74k samples in the whole clean train set.

XLSR. The progress of finetuning the XLSR model on the SADA clean training set is shown in Fig. 2. We can see that even though the validation (eval) loss starts

⁵ Reference (Eng.): “I say, Faisal, aren’t you afraid of delay? Hahaha, you made me laugh. What do you mean? He’s afraid of delay. What is this talk about? In addition to other services, are they afraid of delay? I’m afraid of the discount. Oh, what a pity for us. Just be quiet.”

Prediction (Eng.): “I say, O Fakhn, do you not imagine? Seriously, one made me laugh, O Rabukh, the laughter subsided. What does imagination imagine? What is this talk, this talk, did imagination subsided? What does imagination imagine? [repeated 15 times] What does imagination think? [repeated 8 times] What”.

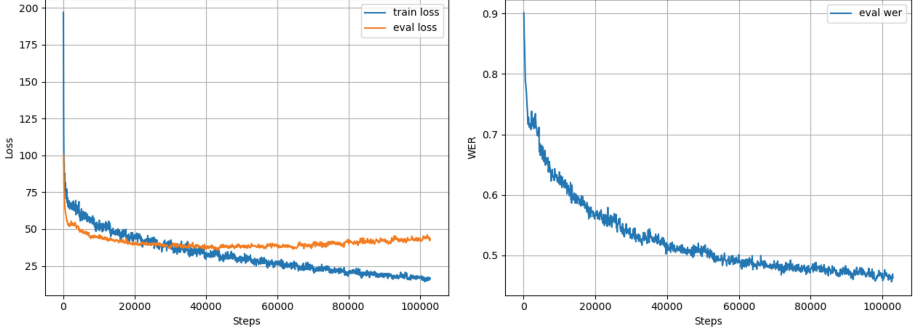


Fig. 2. Finetuning progress of the XLSR model on the SADA clean training set.

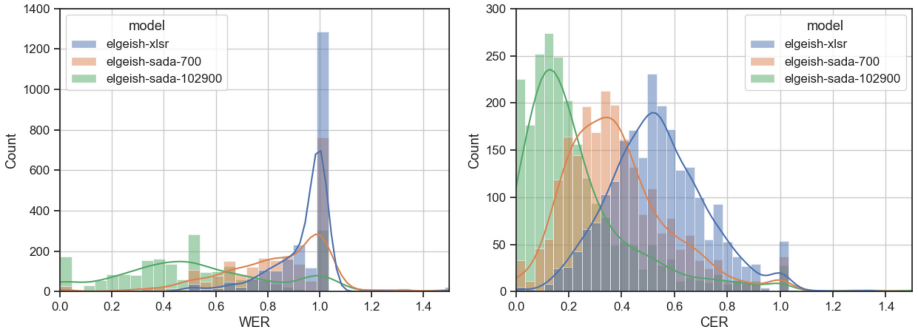


Fig. 3. WER (left) and CER histograms (right) with count on the y -axis for the El-Geish XLSR model on the SADA test clean dataset after finetuning with 700 and 100k steps.

increasing after 43k steps, the validation WER keeps decreasing. We can also see that there is still room for improvement after 100k steps of finetuning.

The results for the finetuned XLSR model show a substantial improvement in performance, with WER going from 93.75% down to 54.3% and CER from 53.91% down to 23.9%, which is an improvement of over 40%. We can see the changes of the distributions of the WER and CER with finetuning in Fig. 3. The plots also show the performance at 700 steps of finetuning, which we can see for the CER is half way to the final distribution for 100k steps of finetuning, exemplifying the law of diminishing returns.

In Fig. 4 we break down the WER and CER performance of the finetuned XLSR model w.r.t. the sample length and the dialect. We can see that the model performs better on longer samples and struggles with samples shorter than 3s. We can also see that finetuning improves this performance, less so for the short samples.

With respect to the dialect, we can see that the original model performs markedly better on MSA than on the Khaliji and Najdi dialects. This can be

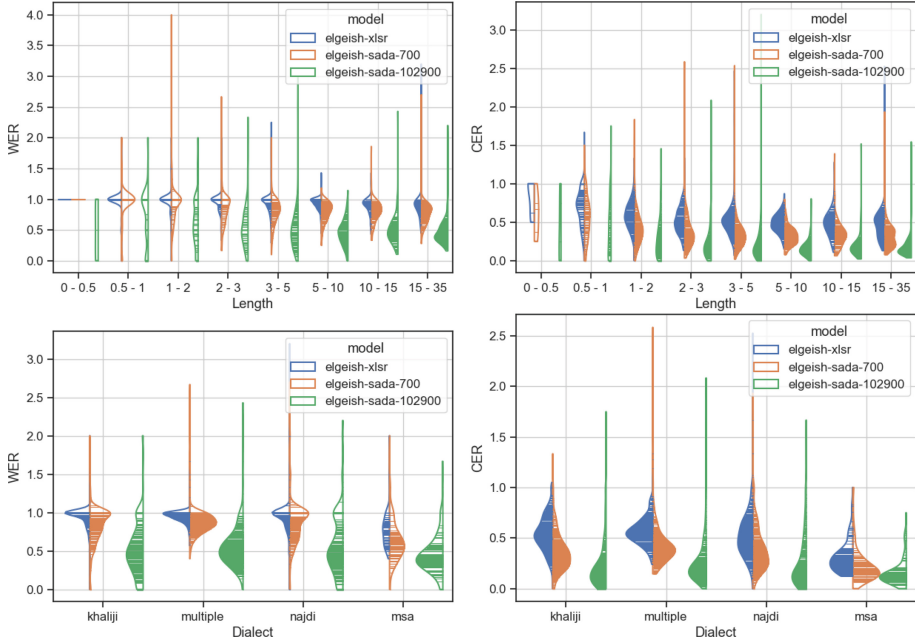


Fig. 4. WER (left column) and CER (right column) for the El-Geish XLSR model on the SADA test clean dataset after finetuning with 700 and 100k steps, spread across sample length (top row) and dialect (bottom row).

expected as El-Geish finetuned XLSR on Damascian via Arabic Speech Database and on MSA via Common Voice. Finetuning significantly improves performance for unseen dialects, with MSA still performing best over all.

This gives a marginal improvement to performance giving a WER of 51.6% and a CER of 22.1%, which is an further improvement of 2.7% WER and 1.7% CER over the finetuned XLSR model.

Finally, finetuning the original XLSR model, instead of the already finetuned El-Geish XLSR model, gives a WER of 91.5% and a CER of 51.4%, which is a slight improvement of 2.3% WER and 1.5% CER over the El-Geish XLSR model, but still worse than the finetuned El-Geish XLSR model. This is probably due to the El-Geish XLSR model being already finetuned for at least 100k steps, meaning that our finetuning brings it to 200k steps, double the number of steps we used for finetuning the original XLSR model.

XLS-R. The loss curves from finetuning XLS-R on SADA are shown in Fig. 5. We can see that the validation loss starts increasing after 20k steps, but the validation WER keeps decreasing.

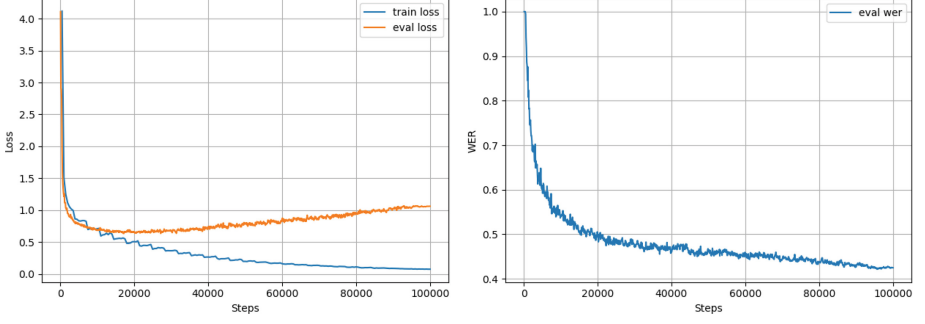


Fig. 5. Finetuning progress of the XLS-R model on the SADA clean training set.

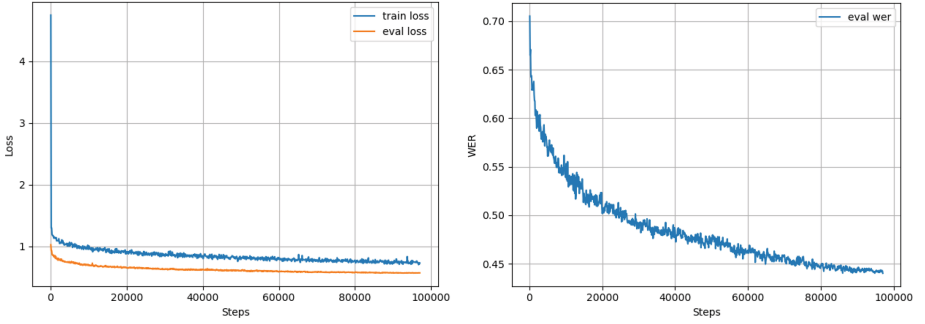


Fig. 6. Finetuning progress of the MMS model on the SADA clean training set.

The finetuned XLS-R model shows worse results than the finetuned on SADA El-Geish XLSR model, with a WER of 90.7% down and CER of 49.2%.⁶

MMS. The finetuning progress of the MMS 1B model on SADA is shown in Fig. 6. We can see that both the validation loss and WER keep decreasing, even though they seem to near convergence. The finetuned MMS 1B model gives a WER of 51.5% and a CER of 19.9%, which is the best performance we achieved on the SADA clean test set with finetuning for all the considered ASR models. The distribution of the WER and CER for the best-performing finetuned models is shown in Fig. 7.

4.3 Language Models

We train a 4-gram language model on the SADA training set using KenLM. The obtained ARPA file is 2.2 million lines long (3 M for a 5-gram model) and a size of 100 MB. We integrate the language model into the XLSR and MMS models

⁶ We also tried continued pretraining of XLS-R on SADA before finetuning, but the contrastive and eval loss curves showed that the process is failing.

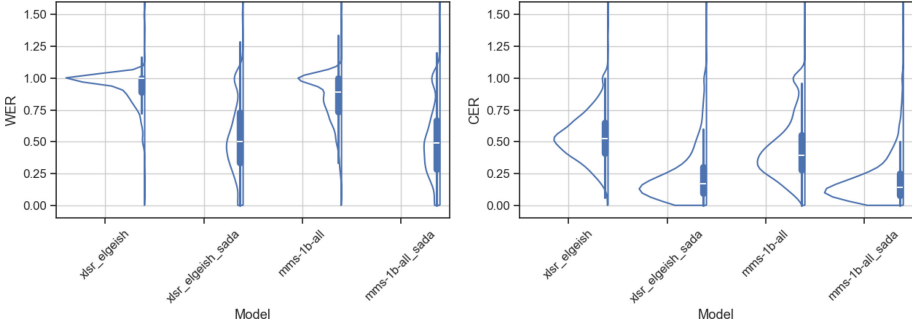


Fig. 7. WER (left) and CER (right) distributions for the El-Geish XLSR and MMS models and their versions finetuned on SADA.

and obtain a absolute WER improvement of 9 and 11% and a CER improvement of 4 and 2%. With this we achieve the best performing ASR system on SADA with a WER of 40.9% and a CER of 17.6% for MMS 1B finetuned on SADA with an LM.

4.4 Noise and Denoising

The results from the evaluation of the XLSR model on the noisy subset are given in Table 3. The results from the evaluation on the clean dataset are added for reference.

Table 3. ASR model performance on SADA in noise and denoising.

Model	Domain	WER	CER
xlsr-elgeish	clean	93.75	53.91
xlsr-elgeish	noisy	96.87	58.27
xlsr-elgeish	denoised	97.29	59.14
xlsr-elgeish +SADA	clean	54.34	23.86
xlsr-elgeish +SADA	noisy	63.43	29.63
xlsr-elgeish +SADA	denoised	66.11	31.58
xlsr-elgeish +SADA (unfreeze)	clean	51.64	22.13
xlsr-elgeish +SADA (unfreeze)	noisy	60.95	28.09
xlsr-elgeish +SADA (unfreeze)	denoised	64.46	30.47
xlsr-elgeish +SADA-noisy (unfreeze)	noisy	58.95	27.10
xlsr-elgeish +SADA-denoised (unfreeze)	denoised	59.13	28.85

We can see that the models, including the ones finetuned on the clean SADA train set, perform worse on the noisy data. The best model obtained with finetuning XLSR with unfreezing the encoder weights xlsr-elgeish +SADA (unfreeze)

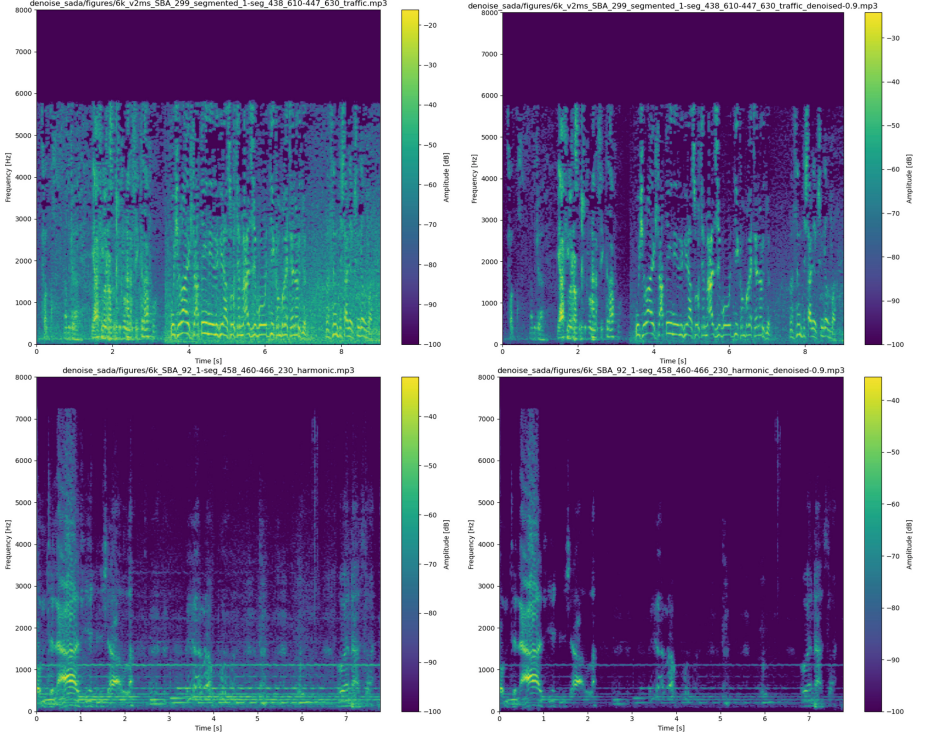


Fig. 8. Example denoising results for traffic noise (top row) and harmonic noise (bottom row).

gives a WER of 60.9% and a CER of 28.1% on the noisy data, which is 9.3% worse WER and 6% worse CER than on the clean data. Finetuning the model on the noisy SADA train set does improve performance and gives a WER of 58.9% and a CER of 27.1%, which is the best performance on the noisy data. We note that this is still much worse than the performance on the clean data.

We next apply the Noisereduce denoising algorithm to the noisy SADA data. Example results of the denoising process can be seen for traffic and harmonic noise in Fig. 8. We set the noise reduction coefficient to 0.9, i.e.90%, to avoid severely damaging the speech signal integrity.

From Table 3 we can see that denoising the noisy data does not help the XLSR model performance. In fact, it is always detrimental to the performance of the system. Even the best performing model – xlsr-elgeish +SADA (unfreeze), obtains 3.5% worse WER and 2.4% worse CER than on the noisy data. Fine-tuning the model to the denoised SADA dataset does improve performance, but it is still worse than finetuning and evaluating the model on the noisy data sans denoising.

Finally, we plot the absolute improvement of CER due to denoising w.r.t. the CER and the Noisereduce average relative noise spectrogram energy in Fig. 9.

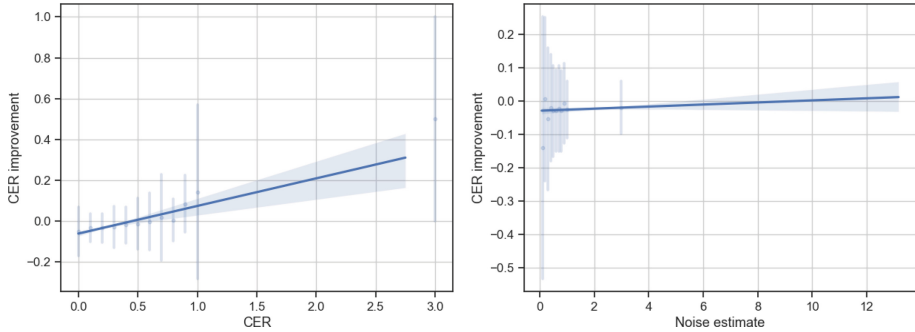


Fig. 9. Absolute improvement of CER due to denoising w.r.t. the CER (left) and the Noisereduce average relative noise spectrogram energy (right).

We bin the data for clearer visualization and compute the average CER and noise estimate for each bin. We fit a regression line to the data and compute the Pearson correlation coefficient, which equals 0.28 for the CER and 0.03 for the noise estimate.

We can see that denoising helps for higher CER, and hurts performance for lower CER. On average it leads to CER improvement if the CER is above 50%. The same is true for the noise estimate, i.e. denoising helps for higher noise.

5 Conclusion

We explored the performance of state-of-the-art ASR models on the SADA Arabic dataset. We found that the 1B parameter MMS model finetuned on SADA with a 4-gram language model gives the best performance with a WER of 40.9% and a CER of 17.6%. The 300M parameter XLSR model finetuned on SADA with an unfrozen encoder and a 4-gram language model gives a competitive WER of 42.6% and a CER of 18.3%, offering a good trade-off between performance and model size. We also explored the impact of noise on the performance on the smaller XLSR model. We found that it performs worse on the noisy data, and that finetuning it on the noisy data is the best path towards improving performance. Denoising the noisy data mostly does not help performance. Although, it does have the potential to improve performance on data with pronounced noise levels.

Acknowledgments. This study was funded by Pi School, Rome, Italy. The authors would like to thank the Pi School for providing the necessary resources and support for this research.

References

1. Abdelhamid, A.A., Alsayadi, H.A., Hegazy, I., Fayed, Z.T.: End-to-end Arabic speech recognition: a review. In: Proceedings of the 19th Conference of Language Engineering, pp. 26–30 (2020)
2. Alharbi, S., et al.: SADA: Saudi audio dataset for Arabic. In: ICASSP 2024–2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 10286–10290. IEEE (2024)
3. Ardila, R., et al.: Common voice: a massively-multilingual speech corpus. LREC (2020)
4. Babu, A., Wang, C., Tjandra, A., Lakhotia, K., Xu, Q., Goyal, N., Singh, K., Von Platen, P., Saraf, Y., Pino, J., et al.: XLS-R: Self-supervised cross-lingual speech representation learning at scale. arXiv preprint [arXiv:2111.09296](https://arxiv.org/abs/2111.09296) (2021)
5. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. Adv. Neural Inf. Process. Syst. **33**, 12449–12460 (2020)
6. Besdouri, F.Z., Zribi, I., Belguith, L.H.: Arabic automatic speech recognition: challenges and progress. Speech Commun. **163**, 103110 (2024)
7. Chowdhury, S.A., Hussein, A., Abdelali, A., Ali, A.: Towards one model to rule all: Multilingual strategy for dialectal code-switching Arabic ASR. arXiv preprint [arXiv:2105.14779](https://arxiv.org/abs/2105.14779) (2021)
8. Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: Unsupervised cross-lingual representation learning for speech recognition. arXiv preprint [arXiv:2006.13979](https://arxiv.org/abs/2006.13979) (2020)
9. Dhoubi, A., Othman, A., Ghoul, O., Khribi, M.K., Al Sinani, A.: Arabic automatic speech recognition: a systematic literature review. Appl. Sci. **12**(17), 8898 (2022)
10. Gulati, A., et al.: Conformer: Convolution-augmented transformer for speech recognition. arXiv preprint [arXiv:2005.08100](https://arxiv.org/abs/2005.08100) (2020)
11. Halabi, N., et al.: Arabic speech corpus. Oxford Text Archive Core Collection (2016)
12. Heafield, K.: KenLM: Faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, pp. 187–197. Association for Computational Linguistics, Edinburgh (2011). <https://www.aclweb.org/anthology/W11-2123>
13. Heafield, K., Pouzyrevsky, I., Clark, J.H., Koehn, P.: Scalable modified Kneser–Ney language model estimation. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, vol. 2: Short Papers, pp. 690–696. Association for Computational Linguistics, Sofia (2013). <https://www.aclweb.org/anthology/P13-2121>
14. Khan, H.I., Abid, A., Moussa, M.M., Abou-Allaban, A.: The Tarteel dataset: crowd-sourced and labeled Quranic recitation (2021)
15. Povey, D., et al.: The Kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding, Hawaii, vol. 1, p. 5-1 (2011)
16. Pratap, V., et al.: Scaling speech technology to 1,000+ languages. arXiv (2023)
17. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning, pp. 28492–28518. PMLR (2023)
18. Sainburg, T., Thielk, M., Gentner, T.Q.: Finding, visualizing, and quantifying latent structure across diverse animal vocal repertoires. PLoS Comput. Biol. **16**(10), e1008228 (2020)

19. Vaswani, A., et al.: Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30** (2017)
20. Wang, Y., Alhmoud, A., Alqurishi, M.: Open universal Arabic ASR leaderboard. arXiv preprint [arXiv:2412.13788](https://arxiv.org/abs/2412.13788) (2024)
21. Young, S., et al.: The HTK book. Cambridge Univ. Eng. Depart. **3**(175), 12 (2002)

Speech Processing for Under-Resourced Languages



Effect of Increased Temporal Resolution on Speech Recognition for French Quebec Using Features from Speech Self-supervised Learning Models

Vishwa Gupta and Gilles Boulianne^(✉)

Centre de Recherche Informatique de Montréal (CRIM), Montreal, QC, Canada
`{vishwa.gupta,gilles.boulianne}@crim.ca`

Abstract. In this paper, we describe advances made in transcribing speech from witnesses and judges in Bastarache and Charbonneau commissions in Quebec. This is a rich audio with many speakers speaking fluently in conversational style. We extract features from SSL models to fine tune hybrid HMM/DNN models and also end-to-end models to compare word error rate (WER) on Bastarache and Charbonneau commission test sets. We also try SSL features extracted at both 10 ms and 20 ms temporal resolution for comparison. In a previous paper, we showed that we reduce WER for all the 15 low resource languages in OpenASR21 evaluation (with only 10 h of training audio), when we increase the temporal resolution of feature parameters computed from the speech SSL models from 20 ms to 10 ms. In this paper, we experiment with Quebec French data with 10 ms and 20 ms temporal resolution. For a training set of 472 h, we show that we still benefit from increasing the temporal resolution from 20 ms to 10 ms. Also, hybrid DNN/HMM models give lower word error rate (WER) than end-to-end speech recognition with 472 h of training audio. With a training set over 1000 h of audio, the end-to-end ASR system gives similar WER as the hybrid DNN/HMM system, and there is no significant improvement with increasing temporal resolution. We also compare our results with Whisper, an automatic speech recognition (ASR) system trained on 680,000 h of multilingual and multitask supervised data collected from the web.

Keywords: French quebec asr · Low-resource · Speech recognition · SSL models · Temporal resolution · Whisper

1 Introduction

In a previous paper [5] we showed that we can reduce the word error rate (WER) significantly for the 15 OpenASR21 low resource languages [9] by increasing the temporal resolution of features from self-supervised learning (SSL) models from 20 ms to 10 ms. In OpenASR21 evaluation the training audio was limited to 10 h for each language. We did not address many issues in that paper [5]. For example, will we get the same WER reductions with increased temporal resolution

when much more training data is available? Also, with 10 h of training, the performance of end-to-end ASR systems is poor, that is why we used DNN/HMM ASR systems for OpenASR21 evaluation. The question is whether end-to-end ASR systems with input features from SSL models give lower WER than the DNN/HMM ASR systems with input features from SSL models.

We used French data we have collected at CRIM from Quebec TV broadcasts, from respeakers of TV programs in Quebec, from transcripts of proceedings of Bastarache and Charbonneau commissions in Quebec, and from Esther [2], Etape [4] and Repere French data [3] from France in order to answer some of these questions.

Both Bastarache and Charbonneau commissions were appointed by Quebec to investigate certain allegations and contain fluent speech in Quebecois French accent. The transcripts of audio were aligned with the audio at CRIM for speech recognition purposes. Overall there was 491 h of audio. We used 472 h of audio for training and 19 h of audio for test. We used this scenario to compare WER for DNN/HMM systems versus end-to-end systems with significantly more training data than the 10 h of training data used in OpenASR21 evaluation. Even with this increased training set, we found that the WER with the DNN/HMM system was lower than that with the end-to-end system.

We then added the rest of the French data (about 1000 h) to the training set. Note that this data is primarily from TV broadcasts in Quebec and France so the speaking style is more controlled. We found that with this added data, the WER with end-to-end systems was similar to WER with DNN/HMM systems.

Another issue was language modeling. With 3-gram based language models, the performance does not necessarily improve when we add more text. The text has to be relevant to the test set. Otherwise, we need to interpolate the LM from training text with the LM from outside text to optimize perplexity. Also this has to be done every time the test sets are derived from different contexts: for example, from commissions data versus TV broadcasts. In the case of end-to-end speech recognition, we used transformer-based language model (LM). With this LM, adding even out-of-domain text reduced WER consistently. With DNN/HMM based system, every module needs to be carefully optimized, while we only need to optimize the size of the end-to-end models depending on the amount of data.

We also used Whisper [12] to transcribe the test sets in order to compare WER obtained by models trained with SSL features versus WER from large pretrained models (like Whisper) available online. Whisper is trained from multi-lingual 680,000 h of audio (both supervised and semi-supervised) and is used for transcription of audio from many different languages without any further training.

2 Dataset

We used French data we have collected at CRIM from Quebec TV broadcasts, from transcripts of Bastarache and Charbonneau commissions in Quebec, from

Esther, Etape and REPERE French data from France, and closed-captioning data from TV broadcasts in Quebec in order to answer some of the questions in the Introduction Section. Table 1 summarizes the various data subsets derived from these sources that will be used throughout this work.

The ETAPE training data consisted of training data for ESTER 2 [2] and the more recently transcribed audio for ETAPE [4] for a total of 300 h of audio. The REPERE training data [3] consists of 47 h of audio from the REPERE corpus. We also had 178 h of internally transcribed audio from French TV broadcasts in Quebec, and some transcribed audio from shadow speakers used for closed captioning of TV broadcasts in Quebec. We removed 12 speakers from this closed-captioning data and created a separate development set we call closed-captioned TV broadcasts (CCTVB in Table 1). Overall, we had 525 h of transcribed audio from French TV broadcasts in Quebec, France and Morocco to form **TVTrain** in Table 1. The audio was down-sampled to 8 kHz and then up-sampled to 16 kHz in order to be able to recognize both TV broadcasts and telephone bandwidth speech. Note that all development and test sets used for evaluation contain only French from Quebec, while the training data includes other varieties.

We also downloaded audio from Bastarache¹ and Charbonneau commissions² set up by the province of Quebec to investigate some allegations. These commissions were headed by a Judge and the various testimonies were recorded. Each recorded session could be a few hours long. These testimonies were also available in transcribed pdf format. But these testimonies were not transcribed verbatim. The transcripts contained speaker information and other non-spoken content.

The processing involved in transcribing the Bastarache and Charbonneau commission data is described in [13]. In this paper the results reported use the version 1.0.0 of the corpus described in [13] except for results in Sec 4.5 which use the latest version.

The Bastarache and Charbonneau commission audio version 1.0.0 is 491 h long. The Bastarache audio is 80.2 h long and was divided into 72 h of training set (45 speakers), 4 h of female development set (7 speakers), and 4.2 h of male development set (8 speakers). The speakers in training and development sets did not overlap. The Charbonneau audio is 411 h in duration and was divided into 400 h of training set (461 speakers), 5 h of female development set (17 speakers) and 5.8 h of male development set (18 speakers). Together Bastarache and Charbonneau training sets total 472 h (**BasCharTrain** in Table 1).

The speakers in development sets were chosen towards the tail end of the duration in order to maximize the total number of speakers and to avoid any speaker to dominate the results. This should lead to less biased results. Also, speakers with generic names were not considered for the development sets. Bastarache and Charbonneau development sets together contains 9 h and the test sets 10 h (**BasChar** in Table 1).

¹ https://fr.wikipedia.org/wiki/Commission_d%27enquête_sur_le_processus_de_nomination_des_juges_du_Québec.

² https://en.wikipedia.org/wiki/Charbonneau_Commission.

The context of the text from TV broadcasts (mostly from news, sports and weather) is quite different from the Bastarache and Charbonneau commissions text that is related to the issues the commissions were investigating. The speaking style in TV broadcasts is mostly read speech while the commissions data is mostly testimonies by speakers in fluent French which may contain hesitations, pauses, repetitions and other characteristics of conversational speech. To accommodate the different contexts, we created 5 development sets: one male and one female development set for Bastarache commission, one male and one female development set for Charbonneau commission, and one development set with 12 shadow speakers from TV broadcasts. This way, we can see any male/female bias, and also any bias in the decoding between TV broadcasts and commissions data.

Table 1. Designation of the various datasets used in this work.

Designation	Role	Source	Size
TVTrain	ASR training	TV broadcasts from CRIM, Esther, Etape, REPERE	525 h
BasCharTrain	ASR training	Public inquiries	472 h
TVBasCharTrain	ASR training	TVTrain + BasCharTrain	1000 h
BasChar	Dev. and test	Public inquiries	9 h and 10 h
CCTVB	Dev. and test	Closed-captioners	12 speakers
SmallLM	Training text	All transcribed data	16M words
LargeLM	Training text	SmallLM + internet + newspapers	326.5M words

3 ASR Approach with SSL Features at Different Temporal Resolutions for Hybrid DNN/HMM Versus End-to-End ASR

The idea here is to develop the best possible speech recognition system for French as spoken in Quebec using the data we have available at CRIM. We have already shown in a previous paper [5] that features from speech self-supervised learning (SSL) models give significantly lower word error rate (WER) than MFCCs [1]. We will use the features from SSL models in both hybrid DNN/HMM and end-to-end ASR for comparison. Since end-to-end ASR models require large training data, we will use training audio from just the Bastarache and Charbonneau commissions (BasCharTrain, 472 h), and also from all the training data from Bastarache, Charbonneau, and TV broadcast audio (TVBasCharTrain, over 1000 h) to compare the WER at different training sizes. We also want to see whether mixing audio from different spoken contexts for training will still lead to reduced

word error rates on the various development sets. Overall, we would like to compare performance for different training sizes and for development audio from different spoken contexts.

We also want to compare WER with features extracted from the SSL speech models with temporal resolution of 20 ms versus 10 ms for much larger training audio than the 10 h used in [5]. Here also, to train DNN/HMM system using Kaldi toolkit [10], we extract the features from the SSL models using Transformers³. We extract features at both 10 ms and 20 ms temporal resolution. Separate DNN/HMM systems are trained for 20 ms and 10 ms frame intervals.

Section 3.2 in reference [5] shows how to obtain features with 10 ms frame interval from the SSL speech models that provide features at 20 ms frame interval, without re-training the SSL models. We extract features from the original audio, and from the same audio trimmed by 10 ms in the beginning (trimming the beginning of the audio by 10 ms advances the features of the audio by 10 ms compared to the original audio). We then combine the two sets of features by concatenating frames alternately (interleaving them in time) to give a new set of features with 10 ms frame interval.

Note that the total number of frames for features with 20 ms frame interval is half of that with 10 ms interval. For that reason, we create a 20 ms frame interval dataset by concatenating the set of features from the trimmed audio and the set of features from the original audio, but without interleaving the individual frames, resulting in twice the total number of frames but still a 20 ms frame interval.

3.1 Training the Hybrid DNN/HMM System

For training the DNN system, we used the Kaldi toolkit [10]. We train a factored time delay neural network (TDNN-F) [11] for both 10 ms and 20 ms temporal resolution. The features with 20 ms temporal resolution were extracted from the w2v-bert-2.0 SSL model using Transformers. We choose w2v-bert-2.0 SSL model as it was trained from 4.5 million hours of audio, and it gave good WER for all the 15 OpenASR21 languages [5]. Features with 10 ms frame interval are extracted without retraining the SSL model as outlined in Sect. 3.2 of the paper [5]. Training the TDNN-F system requires generating alignment and lattices for the acoustic data. The alignment and lattices are generated using a GMM/HMM system trained with the same acoustic data using 13-dimensional perceptually weighted linear prediction (PLP) features [6]. The HMM/GMM are computed separately for PLP features with 20 ms and 10 ms temporal resolutions. Note that PLP features with 20 ms and 10 ms temporal resolutions are computed with different window sizes in order to ensure reasonable speech overlap between consecutive windows. Window size for 20 ms temporal resolution is 40 ms, while the window size for 10 ms temporal resolution is 25 ms. We tried different model architectures for the two sets of features to get the lowest possible WER.

³ <https://huggingface.co/docs/transformers/en/index>.

For the 10 ms temporal resolution feature, we tried two different model architectures: two streams and one stream, which gave similar results. The 2-stream model architecture is shown in Fig. 1. Each TDNN-F layer is 512 dimensional with a bottleneck dimension of 80. The total number of parameters are 20.4 million.

The model architecture for the 20 ms temporal resolution has 2-streams with 13 TDNN-F layers per stream, stream 1 has time-stride of 2, and stream 2 has a time stride of 4. Because of frame advance of 20 ms in each utterance, the `frame_subsampling_factor` is set to 1 (instead of 3). The time-strides of the 2 streams are also different from Fig. 1 because of 20 ms temporal resolution (instead of 10 ms temporal resolution). Also, the total number of TDNN-F layers are reduced to 13 (from 15). These changes give a significant reduction in WER for SSL features with 20 ms temporal resolution. The model has 20.5 million parameters.

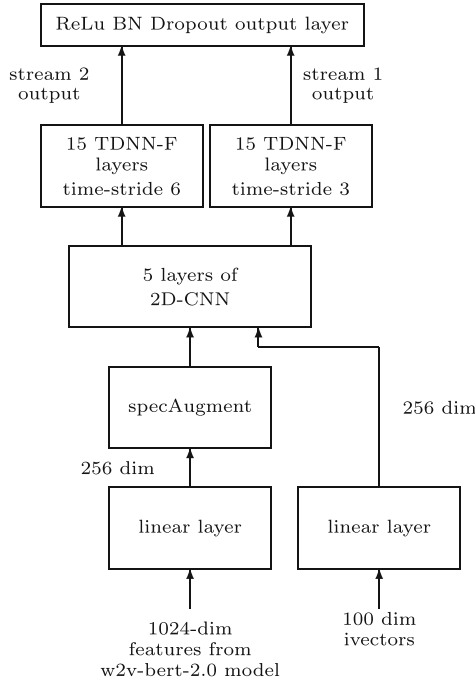


Fig. 1. 2-stream TDNN-F acoustic model with 1024-dimensional features from 24th layer of w2v-bert-2.0 model as input, together with 100 dimensional i-vectors.

3.2 Training End-to-End ASR System

For training end-to-end ASR systems, we used ESPNET toolkit [14] to train the acoustic and language models. We used the setup that gave very good results for

LibriSpeech “test other” test set with features from SSL models. ESPNET has scripts to compute features from pretrained SSL models through s3prl toolkit. So any pretrained SSL model that can be accessed through s3prl toolkit⁴ can be used in ESPNET. We could not find a way to access w2v-bert-2.0 SSL model through the s3prl toolkit in ESPNET, so we used the XLS_R-300m model pretrained on 436k hours of multilingual unlabeled speech. Features from this model gave comparable WER in the OpenASR21 scenario [5].

We used the training and decoding script in the “egs2/librispeech/asr1” subdirectory which used features from WavLM-large SSL model to train an end-to-end conformer acoustic model (see conf/tuning/train_asr_conformer7_wavlm_large.yaml) and Transformer LM (see conf/tuning/train_lm_transformer2.yaml) that gave 3.7% WER on librispeech “test other” test set. We adapted the two configuration files to use XLS_R-300m SSL model and reduce the model sizes in order to be able to train the acoustic and language models with four 80 gigabyte GPUs in one node. We tried different acoustic and language model sizes in order to see how the model size affects WER.

ESPnet adopts hybrid CTC/attention end-to-end ASR [15], which effectively utilizes the advantages of both architectures in training and decoding. The overall model architecture for the acoustic model is shown in Fig. 2. In Fig. 2, the 1024-dimensional features from the XLS_R-300m model are input to a specAugment layer. A linear layer converts the specAugment output to 80-dimensional features. These 80 dim features are followed by two Conv2d subsampling layers, followed by a linear layer with output dimension of 512. This is followed by multiple conformer encoder layers. The output of conformer encoder goes to transformer decoder with multiple decoder transformer layers. The total loss is a combination of attention-based cross entropy loss (weight 0.7) and CTC loss (weight of 0.3) [14] to optimize 5000 dimensional output (size of BPE or byte pair encoding model).

For language modeling, we use a BPE (Byte Pair Encoding) based language model with characters as sub-word units. The size of the BPE model is set to 5000. These units are input and output to a transformer based language model. We trained two different language models: one small transformer model with 64 dimensional embedding layer and 8 encoder layers. Each encoder attention layer is 256 dimensional with 8 heads, with 1024 dimensional feed forward layer. The medium transformer model has 128 dimensional embedding layer and 8 encoder layers. Each encoder attention layer is 512 dimensional with 8 heads, with 2048 dimensional feed forward layer. Both the models have a decoder layer with 5000 outputs corresponding to the size of the BPE model.

During decoding, ESPnet performs joint decoding by combining attention-based and CTC-based scores with the log probability from the transformer based language model in a one-pass beam search algorithm [7].

⁴ <https://github.com/s3prl/s3prl>.

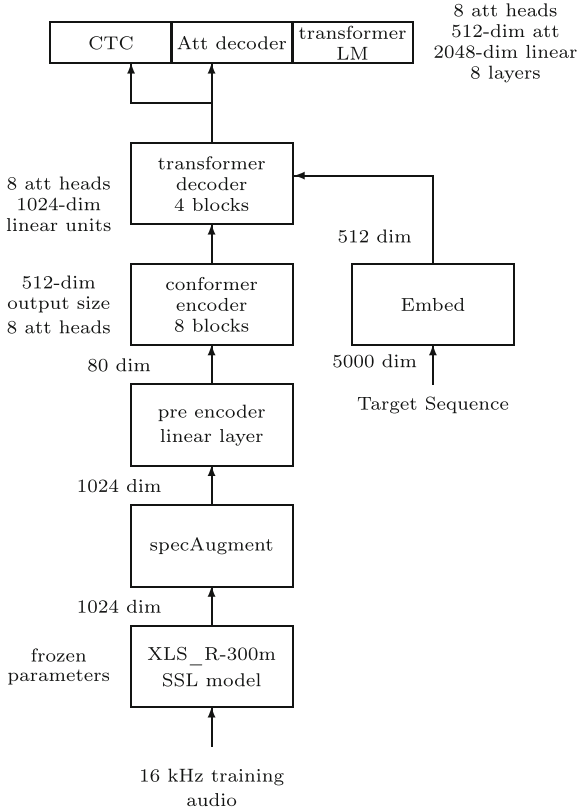


Fig. 2. End-to-end acoustic model architecture. The dimensions correspond to the large acoustic model that we used with 66.25 M trainable parameters (not including the frozen SSL model parameters). The transformer language model dimensions as shown are for the biggest LM model we used with 28.49 M parameters.

4 Experimental Results with Different Temporal Resolutions

As mentioned in the Introduction Section, the idea is to compare word error rate (WER) for the hybrid DNN/HMM based recognizer with input from SSL model at two different temporal resolutions (10 ms versus 20 ms) with different amounts of training data. We also compute WER with an end-to-end ASR system with different size acoustic and language models as outlined in Sect. 3.2.

We ran several experiments in order to compare (WER) with different training sizes for both acoustic and language modeling. For acoustic modeling, we used three different training datasets: 1. Just from TV broadcasts (TVTrain). 2. Bastarache and Charbonneau commissions data (BasCharTrain), and 3. All the audio data combined (TVBasCharTrain).

For language modeling with TDNN-F acoustic models, we trained 3-gram and LSTM-LM language models with 2 different text datasets: **SmallLM** with transcripts from all the French transcribed training data (16 million words of text), and **LargeLM** which adds text downloaded from internet and text from various newspapers like *La Presse*, etc.) with 326.5 million words (see Table 1). With the smaller **smallLM** text, we trained a small LSTM LM (2-layer LSTM language model using the recipe in Kaldi swbd egs⁵ with reduced cell and embedding dimension of 256). While with the **largeLM** text we trained a large LSTM LM (with cell and embedding dimension of 512) in the hope that a larger LSTM LM will reduce the WER significantly. The LSTM LM is used to rescore lattices generated by decoding with the 3-gram LM (in forward direction followed by backward direction). The best weighting for the LSTM LM is between 0.4 and 0.5 for interpolation with the 3-gram LM.

4.1 Results with DNN/HMM Acoustic Models

The first issue was to decide which SSL models to use for extracting features from the audio. In [5] we compared different SSL speech models for 15 languages in OpenASR21 evaluation. We found that w2v-bert-2.0 and XLS_R-2b models gave very good results. So we chose from that comparison w2v-bert-2.0 models as they are much smaller and will reduce computing and GPU memory for extracting features and training models with a large amount of acoustic data.

Since all the data we are working with is French, we also looked for SSL speech models trained with French data only. We found that LeBenchmark/wav2vec2-FR-7K-large models [8] (available on Hugging Face) were pretrained on wav2vec 2.0 SSL models with 7k hours of French audio. So initially we compared WER for 10 ms temporal resolution versus 20 ms temporal resolution using these wav2vec2-FR-7K-large SSL models. We trained DNN/HMM models at both 10 ms and 20 ms temporal resolution using Bastarache+Charbonneau training set. Then we computed WER on Bastarache male/female development sets and Charbonneau male/female development sets and averaged the WER. This averaged WER is from approximately 20 h of audio. The WER for 10 ms frame advance was 17.6% versus 18.2% for frame advance of 20 ms. So even with 472 h of training audio, there is a small reduction in WER with 10 ms frame advance. In a back to back scenario, we trained acoustic models with features from w2v-bert-2.0 SSL models from Bastarache+Charbonneau training data. The averaged WER over Bastarache and Charbonneau development sets was 17.46%. Even though the difference in WER is small (17.46% for w2v-bert-2.0 versus 17.6% for wav2vec2-FR-7K-large), we chose the w2v-bert-2.0 SSL models since these models were trained from 4.5 million hours of multilingual speech, and they probably cluster speech much more accurately for different languages and dialects including French.

⁵ https://github.com/kaldi-asr/kaldi/egs/swbd/s5c/local/rnnlm/\discretionary-run_tdnn_lstm.sh.

Note that the word error rate (WER) is computed differently in Kaldi toolkit versus ESPNET. So we normalized all scoring by using ESPNET to score results from Kaldi using ESPNET scoring. As a common basis, we used “whisper-basic” cleaner in ESPNET to clean both hypothesized and reference transcripts before computing the WER. “whisper-basic” cleaner lower cases all the words and splits the words with (') and removes the quotes. It also splits the words at hyphen (-). This cleaning reduces the WER significantly compared to Kaldi toolkit. So Tables 2, 3, 4, 5, 6 and 7 reflect this consistent scoring across Kaldi and ESPNET.

Table 2 summarizes the results with different acoustic and language model training sets and for different development sets. All the acoustic models are trained with features from the last encoder layer of w2v-bert-2.0 SSL model. We trained three different acoustic models at 10 ms temporal resolution. Features at 10 ms temporal resolution were obtained by combining SSL features from the original audio and from audio trimmed by 10 ms as outlined in Sect. 3.2 of reference [5]. SSL features with 20 ms temporal resolution were obtained from the original audio and also from audio trimmed by 10 ms (see Sect. 3.1) and then the two datasets were combined. This process can be considered as another data augmentation process.

Column 2 in Table 2 shows results with acoustic model trained from all the data from TV broadcasts only (TVTrain). The CCTVB development set is closed-captioning data from TV broadcasts in Quebec and is quite different from the Bastarache and Charbonneau development sets (commissions proceedings versus TV broadcasts) to see how acoustic and language model training affect the WER for different spoken contexts. To compute the WER for the combined Bastarache and Charbonneau development sets, we computed the WER for Batarache male, Bastarache female, Charbonneau male, and Charbonneau female development sets and averaged them.

Table 2. WER (%) for BasChar and CCTVB dev sets with TDNN-F acoustic models trained from 3 different datasets: TVTrain, BasCharTrain and TVBasCharTrain and two different temporal resolutions.

Train set	TVTrain	BasCharTrain	TVBasCharTrain	TVBasCharTrain
Test set	10 ms	10 ms	10 ms	20 ms
BasChar small LM	18.7	12.9	13.3	13.0
BasChar large LM	17.7	12.5	12.8	12.5
CCTVB small LM	7.4	6.5	4.7	4.9
CCTVB large LM	6.5	6.2	4.4	4.6

From Table 2, we can see that the best result (12.5%) for BasChar dev set is with acoustic models trained with BasCharTrain data or with TVBasCharTrain data with large LM. This is probably because both the acoustic model and language model training sets contain data from both Bastarache and Charbonneau training sets. However, acoustic models in column 2 of Table 2 are not trained with either Bastarache or Charbonneau acoustic data. That is why the WER for BasChar development set is 50% worse than the corresponding error in other columns (that include Bastarache and Charbonneau acoustic data in training). We also see that WER for small LM (in rows 2 and 4) is always worse than WER for large LM. For the CCTVB dev set, the best result is with acoustic models trained from all the data and decoded with large LM. Also, with large acoustic model training set TVBasCharTrain, SSL model features with 10 ms temporal resolution give lower WER for CCTVB dev set than features with 20 ms. All these results show that even though SSL model features give significantly lower WER, we still need to train on acoustic data that includes data from the target audience in order to get good performance on it.

4.2 Results with End-to-End Trained Conformer Acoustic Models with Transformer LM

With increasing training data, the end-to-end speech recognition models give decreasing WER in general. So we would like to compare the WER we get with end-to-end speech recognition models (with features from SSL models) with WER from DNN/HMM based speech recognition systems (with features from SSL model for both 10 ms and 20 ms temporal resolution). As outlined in Sect. 3.2, for end-to-end systems, we used the ESPNET toolkit and a configuration file that gave very good results on librispeech data. The librispeech acoustic training data is about the same size (960 h) as our French data (over 1000 h). We tried three different datasets: BasCharTrain with about 472 h of training, TVBasCharTrain with over 1000 h of training, and TVBasCharTrain2 that also includes all the audio trimmed by 10 ms. Note that in end-to-end systems, there is no concept of 10 ms or 20 ms frame advance. Instead the training data is different: TVBasCharTrain or TVBasCharTrain2 which repeats each utterance in TVBasCharTrain after trimming the utterance by 10 ms. Technically, the TVBasCharTrain2 is same as TVBasCharTrain for 20 ms. The SSL model features are from XLS_R-300m model. For language modeling, we used two different datasets: `smallLM` text (16 million words of text), and `largeLM` (326.5 million words in total). These comparisons can show us how the WER changes with increasing training and model sizes.

Table 3 shows results with 3 different acoustic models and two different language models.⁶ As we can see from this Table, the WER reduces significantly

⁶ Note that we could not compute results for large LM with acoustic model from BasCharTrain data because the BPE model for Column 2 was computed from BasCharTrain text only, while columns 3 and 4 used the BPE model trained from TVBasCharTrain text.

Table 3. WER (%) for BasChar dev set and CCTVB dev set for end-to-end acoustic models trained from 3 different datasets: BasCharTrain, TVBasCharTrain, and TVBasCharTrain2 with audio trimmed by 10 ms, and two different language models: with all the training text (small LM, 16M words), and all the training text plus all the downloaded French text (large LM, 326.5M words). The last two rows show results for a small acoustic model trained with all the acoustic training data and a large LM.

Trainset	BasCharTrain	TVBasCharTrain	TVBasCharTrain2
Testset	472 h	1000 h	1000 h
BasChar small LM	15.3	12.8	12.6
BasChar large LM	-	13.1	12.8
CCTVB small LM	16.0	4.9	4.7
CCTVB large LM	-	4.5	4.3
BasChar small acous large LM	-	-	14.7
CCTVB small acous large LM	-	-	5.0

with increasing acoustic training data. The WER is worse with the BasCharTrain set that only includes audio from Bastarache and Charbonneau training sets (Column 2). The WER for both BasChar and CCTVB dev sets is worse than with DNN/HMM models in Table 2 column 3. The WER for both BasChar and CCTVB dev sets is the lowest with acoustic models trained from all the training data plus the 10 ms trimmed acoustic data (column 4).

Note that WER for BasChar dev set is lower with the small LM for acoustic models in columns 3 and 4. However, the WER for CCTVB dev set is lower with the large LM than with the small LM. So in the end-to-end ASR context, the context of the language model training text should be consistent with the context of the development set for achieving lower WER.

In the last two rows of Table 3, we decode with a small acoustic model and large LM. The small acoustic model gives higher WER for both BasChar dev set (14.8% vs. 12.87%) and CCTVB dev set (5.5% vs. 4.8%). A small acoustic model simulates the scenario where a single classification layer is used after SSL model for decoding purposes. A larger acoustic model trained with the SSL model features reduces WER.

Another result we notice from Tables 2 and 3 is that the word error rate for fluent conversational speech (12.5%) is significantly higher than for TV broadcast speech (4.3%).

4.3 Whisper Decoding

We also ran decoding results with Whisper-large from OpenAI [12]. Whisper is an automatic speech recognition (ASR) system trained on 680,000 h of multilingual and multitask supervised data collected from the web. Whisper can decode speech in many languages without any further training. The decoding was done with temperature zero and beam size 10. The results with Whisper decoding are shown in Table 4.

Table 4. WER (%) for BasChar and CCTVB dev sets with Whisper-large, a multilingual pretrained model from OpenAI that can decode speech from many languages without any further training.

Development set	Word error rate
Bastarache male	14.6
Bastarache female	14.8
Charbonneau female	28.9
Charbonneau male	29.6
BasChar (avg)	22.0
CCTVB	9.0
new test set	15.0

If we compare the WER of whisper with the WER’s in Tables 2 and 3, we see that the best Average WER for the Bastarache and Charbonneau commissions dev sets was 12.5% (from Tables 2 and 3), while with Whisper the WER is 22.0%. Similarly, for the CCTVB dev set the best WER is 4.3% (from Tables 2 and 3), while with Whisper the WER is 9.0%. So, with proper tuning of acoustic and language models from in-context data, we can reduce the WER by over 50%. But still, without any tuning from in-context data, whisper performed quite well.

4.4 Results with a New Proprietary Test Set

We would also like to compare the WER for the different recognizers for a test set from a very different context. This is a proprietary test set from a client with a very different application on a laptop. The application context has nothing to do with court proceedings or TV broadcast contexts. The speech is recorded on a laptop. The test set is small (36 utterances from 3 speakers, read speech for command and control) but it is worth seeing how the language model and acoustic model affect this data.

Table 5. WER (%) for a new proprietary test set with TDNN-F acoustic models trained from 3 different datasets: TVTrain, BasCharTrain and TVBasCharTrain and two different temporal resolutions. small LM trained with 16M words, and large LM trained with 326.5M words.

Trainset	TVTrain	BasCharTrain	TVBasCharTrain	TVBasCharTrain
	10 ms	10 ms	10 ms	20 ms
test set	10.1	19.8	7.7	14.0
small LM				
test set	9.2	15.9	7.2	10.1
large LM				

From Table 5 we see that with DNN/HMM based models, we get the best results with large LM and with acoustic models trained from TVBasCharTrain 10 ms. The reason is probably because BasChar train set is noisy and not in-context for the new test set, while both the new test set and TV training sets are clean and probably in context. The WER in Table 6 for end-to-end decoding show similar results. The best result for DNN/HMM system (7.2% WER) is better than for end-to-end system (13.0%).

Table 6. WER (%) for the new proprietary test set for end-to-end acoustic models trained from 2 different datasets: BasCharTrain, and TVBasCharTrain2, and two different language models: small LM trained with 16M words, and large LM trained with 326.5M words.

Trainset	BasCharTrain	TVBasCharTrain2
test set	31.4	-
small LM		
test set	-	13.0
large LM		
test set	-	17.4
small acous		
large LM		

4.5 Results with Revised Bastarache and Charbonneau Data Transcripts

The Bastarache and Charbonneau data was later revised by decoding the audio with the acoustic models trained from the commissions data and then realigning the reference transcripts with the decoded transcripts [13]. We use version 2.0.0 of the revised data. This revised data also contains additional 400 h of

Charbonneau commission data. The Bastarache revised data was divided into Bastarache train, dev, test and charbonneau data into charbonneau train, dev, and test sets. The Bastarache and Charbonneau train sets together with TV train set (TVBasCharTrainRevised) were then used to train new acoustic models as before with the same architecture as described before for both the end-to-end systems and for DNN/HMM based systems. These revised transcripts resulted in significant reduction in WER for both the dev and test sets for Bastarache and Charbonneau. The WER for both the end-to-end and DNN/HMM systems is shown in Table 7. The DNN/HMM systems have a frame advance of 10 ms while the end-to-end system computes features from XLS_R-300 SSL model with a frame advance of 20 ms. The scoring in this table uses “whisper_basic” to clean the reference and the decoded transcripts before scoring. Note that, with the revised data, the WER has gone down significantly from 12.5% to below 6.2%. Also, with over 1400 h of training data, the end-to-end acoustic models give lower WER than the TDNN/HMM models for Bastarache and Charbonneau dev and test sets.

Table 7. Comparison of WER (%) for the new Bastarache and Charbonneau dev and test sets for end-to-end versus TDNN/HMM acoustic models trained from TVBasCharTrainRevised dataset and language model trained from all the training text plus all the downloaded French text (largeLM, 326.5M words).

System	End-to-End	TDNN/HMM
Bastarache Charbonneau dev set avg	5.2	6.3
Bastarache Charbonneau test set avg	6.2	7.4
CCTVB	4.5	4.4
new proprietary test set	14.0	7.7

5 Conclusion

In this paper, we experiment with Quebec French data to see if features from SSL models at 10 ms temporal resolution give better performance than at 20 ms temporal resolution even at much larger training set size (than 10 h used in 15 OpenASR21 languages). For a training set of 472 h, we show that we still benefit from increasing the temporal resolution of SSL features from 20 ms to 10 ms. Also, hybrid DNN/HMM models give lower word error rate (WER) than the end-to-end ASR system even with 472 h of training audio. With over 1000 h of training audio, we see reduction in word error rate with increased temporal resolution for only the TV broadcasts development set. With over

1400 h of training audio, the end-to-end ASR system yields lower WER than the DNN/HMM based ASR system.

We also compare our results with Whisper, an automatic speech recognition (ASR) system trained on 680,000 h of multilingual and multitask supervised data collected from the web. The results on development sets from two different conversation contexts show that by training with in-context audio, we can reduce the word error rate by over 50% compared to Whisper in both these contexts.

Acknowledgments. The authors would like to thank the Ministry of Economy and Innovation (MEI) of the Government of Quebec for their continued support.

References

1. Davis, S., Mermelstein, P.: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE Trans. Acoust. Speech Signal Process.* **28**(4), 357–366 (1980). <https://doi.org/10.1109/TASSP.1980.1163420>
2. Galliano, S., Gravier, G., Chaubard, L.: The ester 2 evaluation campaign for the rich transcription of French radio broadcasts. In: *Interspeech 2009*, pp. 2583–2586 (2009). <https://doi.org/10.21437/Interspeech.2009-680>
3. Giraudel, A., Carré, M., Mapelli, V., Kahn, J., Galibert, O., Quintard, L.: The REPERE corpus : a multimodal corpus for person recognition. In: Calzolari, N., et al. (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pp. 1102–1107. European Language Resources Association (ELRA), Istanbul, Turkey (2012). <https://aclanthology.org/L12-1410/>
4. Gravier, G., Adda, G., Paulsson, N., Carré, M., Giraudel, A., Galibert, O.: The ETAPE corpus for the evaluation of speech-based TV content processing in the French language. In: Calzolari, N., et al. (eds.) *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012)*, pp. 114–118. European Language Resources Association (ELRA), Istanbul, Turkey (2012). <https://aclanthology.org/L12-1270/>
5. Gupta, V.: Advances in openasr21 evaluation with increased temporal resolution for speech self-supervised learning models. In: Karpov, A., Delić, V. (eds.) *Speech and Computer*, pp. 69–81. Springer, Cham (2024)
6. Hermansky, H.: Perceptual linear predictive (PLP) analysis of speech. *J. Acoust. Soc. Am.* **87**(4), 1738–1752 (1990)
7. Hori, T., Watanabe, S., Zhang, Y., Chan, W.: Advances in joint CTC-attention based end-to-end speech recognition with a deep CNN encoder and RNN-LM (2017). <https://arxiv.org/abs/1706.02737>
8. Parcollet, T., et al.: Lebenchmark 2.0: a standardized, replicable and enhanced framework for self-supervised representations of French speech (2024). <https://arxiv.org/abs/2309.05472>
9. Peterson, K., Tong, A.N., Yu, J.: OpenASR21: the second open challenge for automatic speech recognition of low-resource languages. In: *Proceedings of the Interspeech (2022)*
10. Povey, D., et al.: The Kaldi speech recognition toolkit. In: *Proceedings of the ASRU (2011)*

11. Povey, D., et al.: Purely sequence-trained neural networks for ASR based on lattice-free MMI. In: Proceedings of the Interspeech, pp. 2751–2755 (2016)
12. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision (2022). <https://arxiv.org/abs/2212.04356>
13. Serrand, C., Morsli, A., Boulianne, G.: CommissionsQC: a Québec French speech corpus for automatic speech recognition. In: Proceedings of the Interspeech (2025)
14. Watanabe, S., et al.: Espnet: end-to-end speech processing toolkit (2018). <https://arxiv.org/abs/1804.00015>
15. Watanabe, S., Hori, T., Kim, S., Hershey, J.R., Hayashi, T.: Hybrid CTC/attention architecture for end-to-end speech recognition. *IEEE J. Sel. Top. Signal Process.* **11**(8), 1240–1253 (2017). <https://doi.org/10.1109/JSTSP.2017.2763455>



Modeling Intra-word Code-Switching for Karelian ASR

Irina Kipyatkova^(✉) , Kseniia Kiseleva , Mikhail Dolgushin ,
and Ildar Kagiroy

St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS),
14Th Line, 39 199178 St. Petersburg, Russia
{kipyatкова, kiseleva.k, dolgushin.m, kagiroy}@iias.spb.su

Abstract. This paper addresses issues of modeling Karelian-Russian code-switching for automatic speech recognition, with a focus on intra-word code-switching. Due to grammatical differences between Karelian and Russian, and the lack of automatic translation tools for languages in question, standard augmentation methods relying on parallel translated text corpora are difficult to apply. To address these issues, we developed a set of rules specifically designed for generating words with intra-word code-switching, and then augmented the Karelian text by substituting random words with their corresponding generated counterparts. Besides that, we performed linear interpolation of the Karelian language model with the Russian one. We fine-tuned Wav2Vec2.0-large-uralic-voxpathuli-v2 on both Karelian and Russian speech data with the further integration of the developed language model into the system. An evaluation demonstrates significant accuracy improvement: compared to the baseline system without a language model, we achieved relative WER reductions of 11.3% on the development set and 16.6% on the test set.

Keywords: Livvi-Karelian · Code-Switching · Automatic Speech Recognition · Language Modeling

1 Introduction

In linguistics, the term ‘code-switching’ (CS) usually refers to the speaker’s spontaneous transition from one idiom (i.e. a language, a dialect, a sociolect, etc.) to another. CS can occur between sentences, within a sentence, and even within a single word. Generally, automatic speech recognition (ASR) systems supporting CS are significantly more difficult to develop than their monolingual counterparts. The main difficulty is training data scarcity: multilingual speech, being widely represented globally, is poorly attested in existing datasets, especially in the case of low-resourced languages.

Livvi-Karelian (further also – Karelian) is a low-resourced language spoken in the Republic of Karelia in the Russian Federation. Karelian-Russian CS is a wide-spread phenomenon among speakers of Karelian, because they are bilingual without exception. Development of an ASR system supporting Karelian-Russian CS is problematic due to

data scarcity, the morphological complexity of Karelian and Russian, and their differing grammatical structures. Both are languages with a complex morphology, Russian being inflected and Karelian agglutinative. The phenomenon of intra-word CS consists in morphological adaptation of Russian words to the Karelian language by the borrowing of a Russian word base and the addition of Karelian affixes. For example, a speaker can add Karelian affix ‘*an*’ to the Russian word ‘*учи́иуу*’ (‘college’) to convert a word in an accusative case resulting in ‘*учи́иууан*’ (‘*učiliščan*’ in Latin transcription).

In this paper we explore language modeling aimed at improving the accuracy of Karelian-Russian CS speech recognition, with special attention to intra-word CS phenomena. The rest of the paper is organized as follows. In Sect. 2 we give a survey of works related to CS in the context of speech recognition; in Sect. 3 we present the Karelian speech data used in our research; in Sect. 4 we define our approach to text data augmentation and language modeling; Sect. 5 presents the results of experiments on Karelian-Russian CS speech recognition, followed by the conclusion of our work in Sect. 6.

2 Related Work

There exist two main approaches to developing CS-supporting ASR systems [1]. The first approach involves identifying the boundaries of code-switched speech segments and the languages to which they correspond, then processing each fragment using the relevant monolingual speech recognition system. The parameters used for language identification include acoustic features (for example, i-vectors and bottleneck features) [2], lexical features (for example, part-of-speech tags [3]) and trigger words, i.e. words after which code-switching occurs [4]. A joint application of acoustic and language features can also be employed [5]. The second approach involves the use of a multilingual speech recognition system. In this case, the acoustic model (AM) and language model (LM) are jointly trained to cover both languages and speech recognition proceeds without preliminary language identification. In multilingual speech recognition, the alphabet and a set of phonemes must be unified by combining the phone sets of the two languages, or by mapping the phone sets of the two languages, or by merging similar phone sets of the two languages [6].

The advantage of the second approach over the first is that of the absence of a language identification module. However, a multilingual corpus is needed for the second approach. Collecting a training corpus presenting CS is significantly more challenging than collecting monolingual data, particularly concerning the text data required for language model training. CS in text data occurs much less frequently than in speech. Textual transcripts of speech may not be sufficient for training a language model, especially in the case of low-resourced languages.

In order to expand textual data for a code-switched language modeling task, various data augmentation methods can be employed. Among these, partial translation is the most common. It should be noted, however, that textual data augmentation by partial translation can only be performed if there exists an automatic translation system for the languages at issue. For example, word-to-word translation of Mandarin corpus into Taiwanese was performed in [7]. The authors note that the grammar of Mandarin

is similar to Taiwanese, facilitating word-to-word translation. The lowest word error rate (WER) achieved by the authors was 26.02%. In [8] dialectal Arabic-English texts exhibiting CS were obtained through both dictionary-based replacements of random words in Arabic texts with their English counterparts and application of word-aligned parallel sentences in English. In [9] textual data augmentation was performed by random lexical replacements as well as by application of Equivalence Constraint. According to Equivalence Constraint theory, code-switching occurs only where the surface structures of two languages map onto each other, thus implicitly following the grammatical rules of both languages. The lowest values of WER were 55.04% and 47.28% for Kanari and ESCWA speech corpora respectively. In [10], the authors applied a similar method based on the use of grammatical rules as constraints to synthesize new data. In [11] synthetic CS text data were obtained either by replacing individual words with their translations or by combining randomly selected sentences from the original and parallel (translated) corpora. Proximity of word embeddings was used as a constraint to word replacement. It was evaluated by Symmetric Kullback-Leibler Divergence or by Cosine Distance.

Another approach to text data augmentation is neural network (NN)-based text generation. However, it should be noted that a text with code-switching is a prerequisite for training a NN. This approach was applied in [12], where a LSTM-based language model was used for Frisian-Dutch CS texts. The authors achieved WER of 23.5%. Generative Adversarial Networks (GAN) were applied for Mandarin-English CS text generation in [13]. The proposed method resulted in a WER of 22.82% for the LectureSS speech corpus, and 30.00% for the SEAME corpus of spoken speech.

Methods used for monolingual text augmentation can be applied to CS text augmentation as well, among them are random substitution/insertion/deletion of words or symbols, contextual augmentation, etc. [14–16].

An overview of scientific works suggests, that much attention is paid to intra-sentence CS, but modeling of intra-word CS remains under-investigated. In the context of intra-word code-switching, it is worth noting the research presented in [17], and [18] which address the identification of intra-word CS for Arabic-English, German-Turkish, and Spanish-Wixarika language pairs. Investigation of intra-word CS phenomena within dialectal Arabic-English ASR is presented in [8]. The authors proposed an annotation for morphological CS specifically developed to mark Arabic prefixes and suffixes. They found that an end-to-end system outperforms others in recognizing Arabic prefixes, whereas a TDNN-based system demonstrates superiority in recognizing English embedded words. Therefore, they proposed to combine outputs of end-to-end and TDNN-based systems that allowed them to achieve a WER of 30.6%. The limited attention paid to intra-word CS is primarily due to the fact that languages with the complex morphology more subject to intra-word CS. Karelian is an agglutinative language, therefore intra-word CS must be considered when developing an ASR system for it.

3 Karelian Speech Data

The Karelian speech data used in the present research are derived from two speech corpora collected by our research group. The first corpus is AnKaS¹. AnKaS contains annotations for 13 broadcasts of Livvi-Karelian. The original audio data are freely available at the site of The Russian Television and Radio Broadcasting Company (RTR)². AnKaS was described in detail in [19]. The second corpus is a speech corpus KarRusCoS³. This corpus contains recordings of Karelian-Russian annotated CS speech. The KarRusCoS corpus presents recordings of spontaneous Karelian speech from 41 speakers. Within the current research, we used only recordings without background noise and speech overlapping. The essential characteristics of the corpora are presented in Table 1.

Table 1. Corpus metadata.

Corpus features	Value	
	AnKaS	KarRusCoS
Speakers	17 (7 male, 10 female)	41 (16 male, 24 female)
Duration	4.5h	3 h
Utterances	4385	3012
Word occurrences	32037	22355
Unique words	9117	7091
Code-switching rate	1%	28%
Intra-word code-switching rate	<1%	6%

It is worth mentioning that AnKaS mostly contains formal speech samples, thus CS rate is low (about 1%), while KarRusCoS contains spontaneous speech, with a CS rate of about 28%. Therefore, it was AnKaS that was used for training exclusively. Development and test sets were formed from KarRusCoS data, with 10% of the total data for each set. The rest of the data from KarRusCoS were used for training. Speakers in training, development, and test sets were all different. We performed augmentation of the training part by a modification of pitch, speech rate, and simultaneously modified both pitch and speech rate. Consequently, the volume of speech data was increased fourfold.

¹ AnKaS (Database of Annotations of Karelian Speech) can be found at <https://irinakipyatkova.github.io/AnKaS/>

² <https://tv-karelia.ru/kodirandaine-rodnoy-bereg/>

³ KarRusCoS (Speech Database with Karelian-Russian Code-Switching) can be found at <https://github.com/IrinaKipyatkova/KarRusCoS>.

4 Textual Data Augmentation and Code-Switching Language Modeling

Text corpus used for LM training was collected from periodicals in Livvi-Karelian, the open corpus of Vepsian and Karelian VepKar⁴[20], and other freely available text resources. A 3-gram Karelian language model was trained on these data using SRI Language Modeling Toolkit (SRILM) [21]. For LM training a vocabulary of 143K words (each occurred in the texts at list twice) was used. The text corpus, as well as the process of Karelian language model training, are described in detail in [22]. The text corpus contains CS to Russian (mostly proper names and words that have already been borrowed from Russian), but the number of these instances is not sufficient to train a CS-supporting LM. To account for the probability of Russian words appearing in recognized speech, we performed a linear interpolation of the Karelian LM with a Russian LM previously developed for a Russian ASR system [23], which involved Cyrillic-to-Latin conversion. This process resulted in an increase in the number of unique words in the interpolated LM to 287K. Although the resulting LM does not explicitly include n-grams containing both Russian and Karelian words, the Kneser-Ney smoothing method and back-off technique were employed during model development, enabling the re-evaluation of probabilities for unobserved n-grams.

To address intra-word CS, we augmented the text data by automatically generating such words. Based on an analysis of the KarRusCoS corpus, we identified the affixes that are frequently used for the formation of CS words and formulated a set of rules for the morphological adaptation of Russian words to the Karelian language. The identified affixes, along with their frequencies of occurrence in the KarRusCoS corpus, are presented in Fig. 1.

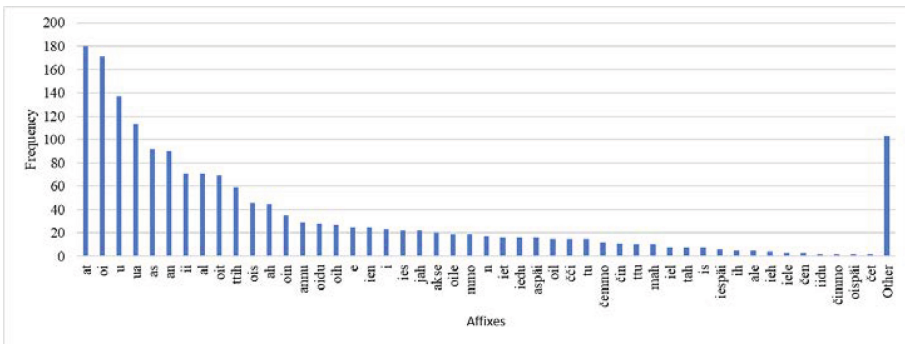


Fig. 1. Distribution of identified affixes and their frequencies.

Some examples of the rules for generating words with the intra-word CS are the following:

1. nouns:

⁴ <http://dictorpus.krc.karelia.ru/en>

- a. formation of the partitive plural:
 - (1) if the stem ends in a soft consonant, add symbol of palatalization (') to the stem;
 - (2) add the ending *oi* to the stem (*очередь* → *очеред'oi* (*očered'oi*));
- b. formation of the inessive singular:
 - (1) if the word is polysyllabic and ends in a diphthong, add *ies* to the stem (*карелия* → *карелиес* (*karelies*));
 - (2) in other cases, if the word ends in a soft consonant, add the ending *is* to the stem (*очередь* → *очередис* (*očeredis*));
 - (3) in other cases, add the ending *as* to the stem (*город* → *городас* (*gorodas*));
2. adjectives:
 - a. formation of the partitive singular:
 - (1) add the ending *oidu* to the stem (*автобусный* → *автобусноиду* (*avtobusnoidu*));
 - b. formation of the inessive:
 - (1) if the stem ends in a soft consonant, add symbol of palatalization (') to the stem;
 - (2) add the ending *ois* to the stem (*автобусный* → *автобусноис* (*avtobusnois*));
3. verbs:
 - a. inflect the word into an imperative form;
 - b. if the imperative ends in a consonant, add *i* to the word in an imperative form;
 - c. for the reflexive verbs, drop the ending *сь* (*s'*) or *ся* (*s'a*);
 - d. formation of the present tense 1st person singular:
 - (1) if the word in imperative form ends in *u* (*i*), add ending *mmo* (*берегу* → *берегуммо* (*beregimmo*));
 - (2) in other case, if the word in imperative form ends in *ü* (*j*) add ending *čemma* (*собирай* → *собирайčemma* (*sobirajčemma*));
 - e. formation of the present tense 1st person plural:
 - (1) if the word in imperative form ends in *u* (*i*) add ending *n*;
 - (2) in other case, if the word in imperative form ends in *ü* (*j*) add ending *čen* (*собирай* → *собирайčen* (*sobirajčen*)).

In total, 33 rules for nouns, 6 rules for adjectives, and 17 rules for verbs were formulated. This set of rules was applied to the vocabulary of 150K Russian words created for Russian ASR during previous studies [23]. After that we added the generated words to the Karelian LM as unigrams. (This expanded vocabulary will henceforth be referred to as the entire vocabulary.) Following that, we augmented the text data with these generated words. For this purpose, we randomly selected 6% of the words in the text dataset. We decided to convert 6% of word based on intra-word CS rate in KarRusCoS corpus. We then determined the grammatical features of these selected words, converted them into their normal form, and translated them into Russian. This procedure was performed using VepKar. If a word could possess different grammatical features or alternative translations into Russian, the variant was chosen randomly. Subsequently, we converted the selected words in the text into words with intra-word CS according to the formulated rules and their grammatical features. When multiple conversions were possible, the utilized variant was chosen randomly. Finally, we trained the LM on the augmented text data. Figure 2 presents a scheme of the proposed language modeling approach.

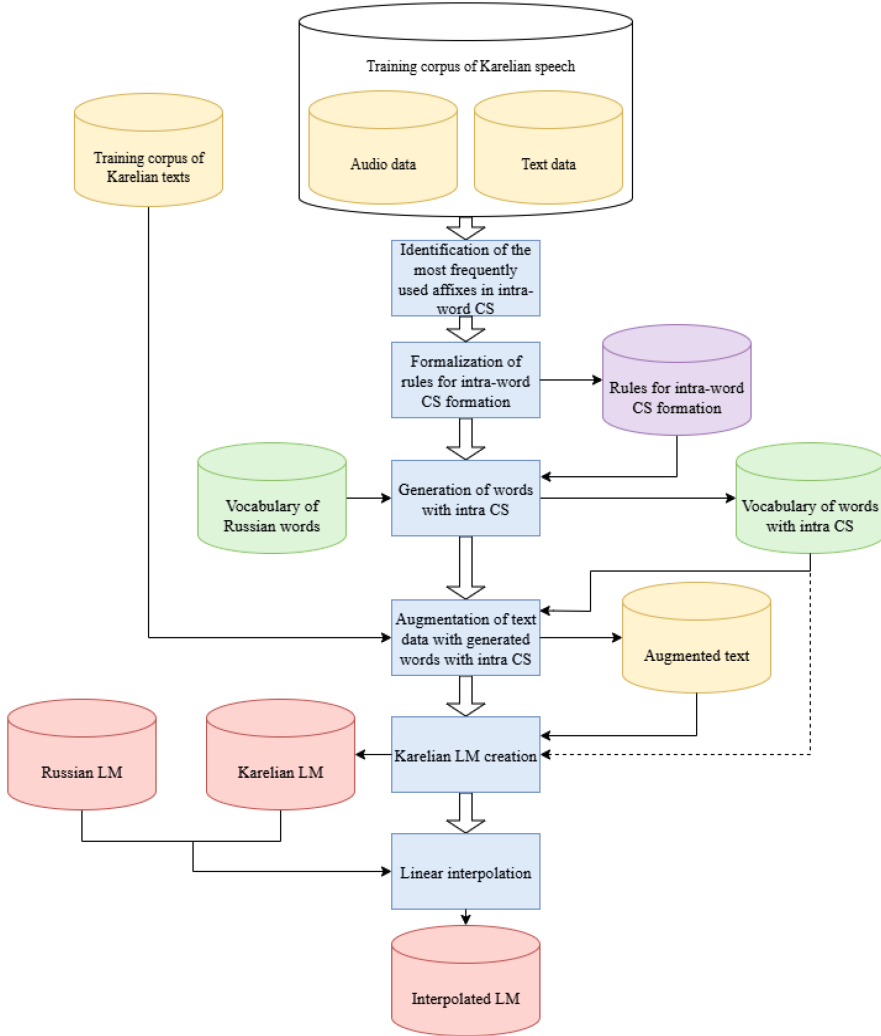


Fig. 2. A scheme of the proposed language modeling approach taking into account CS between Karelian and Russian.

As a result, we trained the following LMs (summarized in Table 2):

1. A LM trained solely on the original Karelian texts (LM1);
2. LM1 linearly interpolated with the Russian LM (LM2);
3. A LM trained on the original Karelian texts with the entire vocabulary interpolated with the Russian LM (LM3);
4. A LM trained on the augmented Karelian texts (comprising only words that occurred in the augmented text), linearly interpolated with the Russian LM (LM4);
5. A LM trained on augmented Karelian texts with the entire vocabulary linearly interpolated with the Russian LM (LM5).

Table 2. Characteristics of the trained LMs.

LM	Training text	Vocabulary	Interpolation with Russian LM
LM1	Original texts	Words from the original texts	No
LM2	Original texts	Words from the original texts	Yes
LM3	Original texts	Entire vocabulary	Yes
LM4	Augmented texts	Words from the augmented text	Yes
LM5	Augmented texts	Entire vocabulary	Yes

The created LMs were evaluated in terms of out-of-vocabulary (OOV) rate and perplexity, both calculated on the transcriptions of the Dev and Test sets of the speech data. The results obtained are presented in Table 3. Besides that, we conducted experiments using different interpolation coefficients. The coefficient specified in Table 3 corresponds to the Karelian model. We have tried different interpolation coefficients. The lowest values of perplexities were obtained with interpolation coefficient equal to 0.8. For comparison, perplexities obtained with interpolation coefficients of 0.7 and 0.9 are presented in Table 3 as well.

Table 3. Perplexities and OOV rates of created LMs.

LM	Vocabulary size, K	Dev set				Test set			
		OOV rate	Perplexity			OOV rate	Perplexity		
LM1	143	21.53	1476.18			20.88	1530.97		
			Interpolation coefficients				Interpolation coefficients		
			0.7	0.8	0.9		0.7	0.8	0.9
LM2	287	11.28	1753.6	1728.0	1812.7	9.93	1972.7	1947.5	2047.4
LM3	851	7.67	2518.3	2480.6	2596.1	7.04	2618.3	2584.2	2712.3
LM4	309	10.11	1939.8	1904.0	1986.4	9.03	2128.8	2095.2	2193.3
LM5	851	7.67	2465.9	2418.8	2518.9	7.04	2572.5	2529.8	2643.6

From Table 3, it can be concluded that the proposed approach leads to a significant decrease in the number of OOV words; however, it also results in an increase in perplexity.

Subsequently, we performed text augmentation in an iterative mode. The procedure for converting random words was carried out multiple times, with a distinct random seed set for each instance. Beginning from the second iteration, only new sentences were added to the text material. The dependencies of the OOV rate and perplexity on the number of iterations are presented in Fig. 3 and Fig. 4, respectively.

Figure 3 illustrates that the OOV rate decreases as the iteration number increases for the Dev set. However, on the Test set, the decrease was not statistically significant.

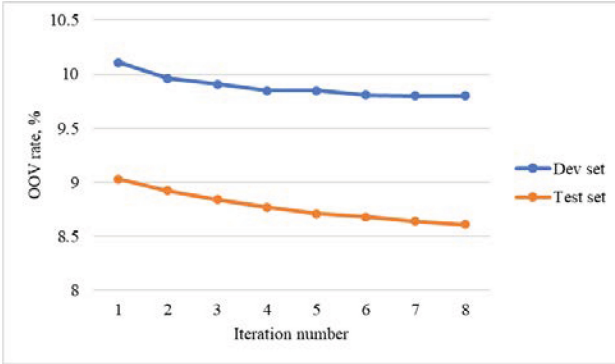


Fig. 3. Dependency of the OOV rate on the number of iterations.

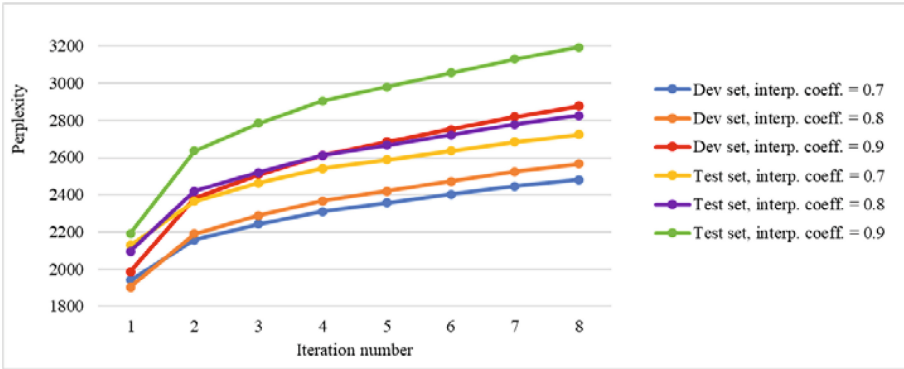


Fig. 4. Dependency of the perplexity on the number of iterations.

This disparity is due to the small size of the datasets and the absence of speaker overlap between them. The perplexity value increases with the number of iterations, which can potentially have a negative impact on speech recognition results. Experiments on Karelian-Russian CS speech recognition are presented in the next section.

5 Speech Recognition Experiments

For our experiments, we used Wav2Vec2.0-large-uralic-voxpathuli-v2 model which is a variation of the multilingual XLSR [24] model developed by Facebook AI Research. Wav2Vec2.0-large-uralic-voxpathuli-v2 has 300m parameters. It is pretrained on 42.5 h of unlabeled speech data of the Uralic languages from the VoxPopuli corpus [25]. We fine-tuned this model on Karelian training set (including augmented data), as well as on 6 h of the Russian speech corpus data described in [26]. The fine-tuning was performed for 10k steps, with a batch size of 8 and 4 gradient accumulation steps. Experimental results, in terms of WER, are presented in Table 4.

The baseline model, without a LM, resulted in a WER equal to 41.47% and 46.38% on the Dev and Test sets, respectively. The use of a LM trained solely on Karelian

text reduced the WER to 41.19% and 44.00%. The most significant improvement was achieved with the LM obtained through linear interpolation of the Karelian and Russian LMs. Further improvement was observed when the model was trained on augmented text, with the best results obtained when the entire vocabulary was utilized as unigrams in this model.

Table 4. Experimental results of Karelian speech recognition experiments.

LM	WER, %					
	Dev			Test		
Without LM	41.47			46.38		
LM1	41.19			44.00		
	Interpolation coefficients			Interpolation coefficients		
	0.7	0.8	0.9	0.7	0.8	0.9
LM2	37.64	37.59	37.55	39.01	39.05	39.13
LM3	37.19	37.09	37.14	38.73	38.75	38.78
LM4	37.15	37.15	37.15	38.80	38.78	38.85
LM5	36.97	36.97	36.94	38.69	38.67	38.71
LM4 (4 iterations)	37.15	37.09	36.90	38.99	38.95	39.04
LM5 (4 iterations)	37.00	36.93	36.76	38.83	38.82	38.83

Subsequently, we conducted experiments with LMs trained on texts augmented over several iterations. The results are presented in Figs. 5 and 6. Increasing the number of iterations led to a decrease in WER on the Dev set, and the lowest value of WER equal to 36.76% was obtained by applying LM with entire vocabulary, which was trained on text data augmented over four iterations and linearly interpolated with the Russian LM using an interpolation coefficient of 0.9. However, increasing the number of text augmentation iterations had no significant effect on the WER for the test set. The lowest WER obtained on the Test set was thus 38.67%, achieved with the application of an entire vocabulary LM trained on text data augmented over one iteration and linearly interpolated with the Russian LM using an interpolation coefficient of 0.8.

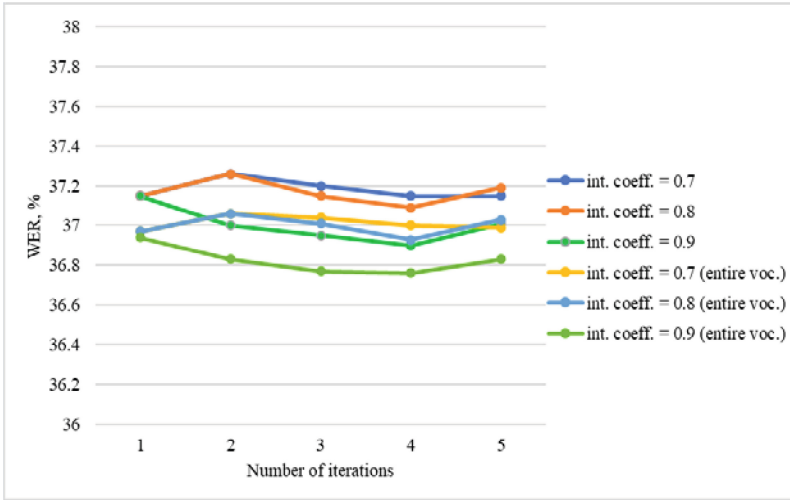


Fig. 5. Dependency of WER on number of interactions (Dev set).

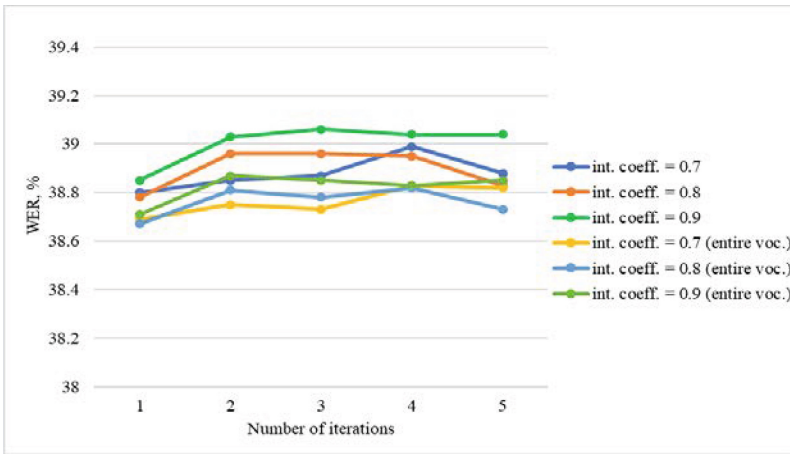


Fig. 6. Dependency of WER on number of interactions (Test set).

6 Conclusions

In the current research we proposed an approach for language modeling for Karelian-Russian CS speech recognition. We apply linear interpolation of Karelian and Russian LMs to include probabilities of Russian words into LM. In order to take into account intra-word CS we proposed a set of rules for artificial generation of such words for text data augmentation.

We have obtained different results for Dev and Test sets, namely increasing the number of augmentation iterations led to reduction of WER on development set, but had no effect when testing on test set. It can be due to small size of the speech corpora

used. We avoid overlapping of speakers between training, development, and test sets. Different speakers may switch to another language differently that results in different improvements achieved for different speakers. However, in our experiments we showed that the proposed approach allows reducing the OOV rate and WER. Relative WER reductions was 11.3% on the development and 16.6% on the test sets compared to the baseline system without LM.

It is important to specifically address the WER obtained. While this metric, as reported within this paper, might seem high, a closer look shows that this result is affected by several important - and inevitable at the present stage of our research - issues. One of the most crucial factors is the scarcity of training data available, which greatly restricts the model's ability for generalizations when encountering word forms which are scarcely represented, or altogether absent within the training data. Moreover, the current Russian-Karelian bilingualism has resulted in a hybrid linguistic system, when a shared grammatical framework is filled in with elements of both of the idioms in quite a chaotic and inconsistent manner. This is the source of numerous irregularities in word formation, such as morpheme variants and deviations from normative morphophonological patterns, all of which increase the WER.

Beyond these issues, a significant source of recognition difficulties arises from the linguistic variability within the speech data due to the natural co-occurrence of multiple Karelian varieties and occasional usage of Finnish. In particular, beyond the Karelian-Russian CS, speakers often switch between different varieties of Karelian, such as Ludic and Karelian Proper. Moreover, even within Livvi-Karelian, being the focus of this study, substantial internal variation can be observed. A good example is the verb form '*zavodimmo*' (as written in the standard modern Livvi orthography), which is pronounced as '*zavodiimmo*' (with a long /i/) in the Kotkatjärvi region. Phonological and morphological variations of this kind, such as alterations in vowel length, stem shape, or suffix usage are inherent features of Karelian dialects and are not fully captured by our current LM and rule-based augmentation procedures. In some cases, speakers may even shift temporarily into Finnish, thus introducing phonological and morphological forms that significantly deviate from those of Livvi, introducing additional recognition issues.

The consequences of this dialectal and even cross-language variations lead to multiple phonetic variants of one and the same lexical item, complicating development of both acoustic and language models. Furthermore, inconsistent representation of these variants in the training corpus directly causes the increase of the OOV rate and further impacts recognition quality, because the model cannot adequately process items not encountered during the learning process.

Taking these issues into consideration, we should conclude that despite the observed high WER metric, the developed system demonstrates a good and robust performance, since all the factors that hinder its accuracy can be attributed to the biased nature of the training data, thus clearly showing a path for future improvements and investigations.

It should be additionally noted that, although a linear interpolation of Karelian and Russian LMs was performed, the resulting LM still does not account for probabilities of switching between Karelian and Russian. Therefore, in the further works, we are going as well to elaborate the rules for an automatic translation from Livvi-Karelian to Russian,

with further text augmentation based on translation of random words from Karelian to Russian.

Acknowledgments. This research was funded by the Russian Science Foundation, grant number 24–21–00276, <https://rscf.ru/en/project/24-21-00276/>

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bhuvanagiri, K., Kopparapu, S.K.: Mixed language speech recognition without explicit identification of language. *Am. J. Signal Process.* **2**(5), 92–97 (2012)
2. Richardson, F., Reynolds, D., Dehak, N.: Deep neural network approaches to speaker and language recognition. *IEEE Signal Process. Lett.* **22**(10), 1671–1675 (2015)
3. Winata, G.I., Madotto, A., Wu, C.S., Fung, P.: Code-switching language modeling using syntax-aware multi-task learning. In: *Proceedings of the Third Workshop on Computational Approaches to Linguistic Code-Switching (CALCS'2018)*, pp. 62–67 (2018)
4. Adel, H., Vu, N.T., Kraus, F., Schlippe, T., Li, H., Schultz, T.: Recurrent neural network language modeling for code switching conversational speech. In: *Proceedings of 2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2013)*, pp. 8411–8415 (2013)
5. Ramanarayanan, V., Pugh, R., Suenderman-Oeft, D.: Automatic turn-level language identification for code-switched spanish–english dialog. In: *Proceedings of 9th International Workshop on Spoken Dialogue System Technology (IWSDS'2019)*, pp. 51–61 (2019)
6. Mustafa, M.B., et al.: Code-switching in automatic speech recognition: the issues and future directions. *Appl. Sci.* **12**(19), 9541 (2022)
7. Hsieh, I. T., Wu, C. H., Wang, C. H.: Acoustic and textual data augmentation for code-switching speech recognition in under-resourced language. In: *IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 302–307 (2020)
8. Hamed, I., Habash, N., Abdennadher, S., Vu, N.T.: Investigating lexical replacements for arabic–english code-switched data augmentation. In: *Proceedings of the Sixth Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2023)*, pp. 86–100 (2023)
9. Hussein, A., Chowdhury, S.A., Abdelali, A., Dehak, N., Ali, A., Khudanpur, S.: Textual data augmentation for Arabic–English code-switching speech recognition. In: *Proceedings of SLT*, pp. 777–784 (2023)
10. Pratapa, A., Bhat, G., Choudhury, M., Sitaram, S., Dandapat, S., Bali, K.: Language modeling for code-mixing: the role of linguistic theory based synthetic data. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, vol. 1: Long Papers, pp. 1543–1553 (2018)
11. Chuang, S.P., Sung, T.W., Lee, H.Y.: Training code-switching language model with monolingual data. In: *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7949–7953 (2020)
12. Yilmaz, E., Heuvel, H., van Leeuwen, D.A.: Acoustic and textual data augmentation for improved ASR of code-switching speech. In: *Proceedings of Interspeech-2018*, pp. 1933–1937 (2018)

13. Chang, C.-T., Chuang, S.-P., Lee, H.-Y.: Code-switching sentence generation by generative adversarial networks and its application to data augmentation. In: *Proceedings of Interspeech-2019*, pp. 554–558 (2019)
14. Şahin, G.G.: To augment or not to augment? A comparative study on text augmentation techniques for low-resource NLP. *Comput. Linguist.* **48**(1), 5–42 (2022)
15. Wan, Z., Wan, X., Peng, W., Li, R.: New datasets and controllable iterative data augmentation method for code-switching ASR error correction. In: *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 8075–8087 (2023)
16. Kobayashi, S.: Contextual augmentation: data augmentation by words with paradigmatic relations. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 2 (Short Papers), pp. 452–457 (2018)
17. Sabty, C., Mesabab, I., Çetinoğlu, Ö., Abdennadher, S.: Language identification of intra-word code-switching for Arabic-English. *Array* **12**, 100104 (2021)
18. Mager, M., Çetinoğlu, Ö., Kann, K.: Subword-level language identification for intra-word code-switching. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1 (Long and Short Papers), pp. 2005–2011 (2019)
19. Kipyatkova, I., Kagirow, I., Dolgushin, M., Rodionova, A.: Towards a Livvi-Karelian End-to-End ASR system. In: Karpov, A., Delić, V. (eds.) *SPECOM 2024, LNAI 15299*, pp. 57–68 (2025)
20. Boyko, T., et al.: The open corpus of the veps and karelian languages: overview and applications. In: *Integration Processes in the Russian and International Research Domain: Experience and Prospects*, pp. 29–40 (2022)
21. Stolcke, A., Zheng, J., Wang, W., Abrash, V.: SRILM at sixteen: update and outlook. In: *Proceedings of 2011 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, p. 5 (2011)
22. Kipyatkova, I., Kagirow, I.: Deep models for low-resourced speech recognition: Livvi-Karelian case. *Mathematics* **11**(18), 3814 (2023)
23. Kipyatkova, I., Karpov, A.: Lexicon size and language model order optimization for Russian LVCSR. In: Zelezny et al. (eds.) *SPECOM 2013, LNAI 8113*, pp. 219–226 (2013)
24. Conneau, A., Baevski, A., Collobert, R., Mohamed, A., Auli, M.: Unsupervised cross-lingual representation learning for speech recognition. In: *Proceedings of Interspeech-2021*, pp. 2426–2430 (2021)
25. Wang, C., et al.: VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In: *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, vol. 1: Long Papers, pp. 993–1003 (2021)
26. Kipyatkova, I.: Experimenting with hybrid TDNN/HMM acoustic models for Russian speech recognition. In: Karpov, A. et al. (eds.) *SPECOM 2017, LNAI 10458*, pp. 362–369 (2017)



Improving Whisper-Based Serbian ASR Using Synthetic Speech

Vuk Stanojev¹ , Tijana Nosek¹ , Siniša Suzić¹ , Darko Pekar² ,
Vlado Delić¹ , and Milan Sečujski^{1,2} 

¹ Faculty of Technical Sciences, University of Novi Sad, Novi Sad, Serbia
vukst@uns.ac.rs

² AlfaNum Ltd., Novi Sad, Serbia

Abstract. In the field of automatic speech recognition (ASR), state-of-the-art results are achieved by end-to-end models. These models are sequence-to-sequence models and are trained using pairs of speech and corresponding texts, which implies that additional finetuning of underlying language models is not possible. In this paper we demonstrate that the performance of Serbian Whisper-based ASR can be improved by leveraging data generation with a high quality text-to-speech (TTS) system in Serbian. Synthetic speech is produced based on text extracted from Serbian web-scale text corpus, SrWAC, using data curation and large language model (LLM)-based normalization to mitigate problems in rendering Serbian pronunciation. A total quantity of 1500 h of speech is generated exploiting 9 text-to-speech voices based on deep-neural architectures and neural vocoding. The experiments are conducted on the medium Whisper model. The baseline model is initially finetuned using 1300 h of transcribed data and then additionally finetuned by synthetic speech, during which process the encoder section of the system is kept frozen. The experimental results confirm the character and word-error rate improvements on the CommonVoice database, as well as on real-life recordings.

Keywords: ASR · Whisper · TTS · fine-tuning · LLM

1 Introduction

The introduction of end-to-end (E2E) models into the field of automatic speech recognition (ASR) has made a great impact [1]. End-to-end ASR models are sequence-to-sequence models trained using pairs of speech and corresponding textual transcriptions, which prevents them from direct language modeling improvements based on additional data in a textual form.

Many end-to-end ASR models use tokens as targets, where tokens may represent words, sub-words, or individual characters. However, once the token inventory is selected, it remains fixed throughout the ASR training process. Since the model is restricted to emitting pre-defined units only, words that never appear in the training transcripts have no dedicated tokens. Assuming that sub-word units are used as tokens,

when such an out-of-vocabulary (OOV) word is spoken at inference time, the system is forced to (i) approximate it by combining the most suitable sub-words, (ii) spell it out character-by-character, or (iii) emit a generic <unk> symbol. While sub-word and character-based systems can theoretically construct unseen words, the absence of certain words in the training data thus leads to poorer recognition accuracy, particularly in case of words that require accurate composition from unfamiliar token sequences. Each of the three fall-backs mentioned above increases the word-error rate, and the errors will be propagated to downstream NLP components. For this reason text-to-speech synthesis (TTS) can be utilized for generating potentially very large quantities of additional training data, which could cover a wide variety of domains and introduce words rarely encountered in their spoken form.

A number of different approaches in combining natural and synthetic speech in ASR training have been proposed. In [2] mel-spectrograms generated by Tacotron2 [3] single-speaker based TTS are used to generate mel-spectrograms and combined with mel-spectrograms extracted from natural speech data in model adaptation. The suggested approach improves out-of-vocabulary word recognition for acoustic-to-word E2E ASR model. In order to overcome data mismatch between real and TTS-generated speech, authors also suggest encoder-freezing procedure, i.e. encoder is copied from the base model and its weights are not updated during adaptation. A similar approach, applied in the adaptation of recurrent neural network transducer (RNN-T) by exploiting commercially available TTS, is presented in [4]. Additionally, it is suggested that applying different types of regularization to the encoder, such as elastic weight regularization [5], instead of just freezing it could also be beneficial. In [6] authors also suggest improving RNN-T model by adaptation with synthetic speech, but also introduce multi-stage training strategy, which includes freezing the LSTM layers of the encoder during finetuning with synthetic speech data and unfreezing all layers in later stages, as well as a regularization loss between frozen and unfrozen model parameters. Since the ASR training should cover as many as possible different speakers, the same authors in [7] suggest incorporating multi-speaker TTS model yielding better results compared to the A2W ASR model adapted using only single-speaker TTS generated speech. Data variation in generated TTS speech is also investigated in [8]. The same speaker data is used to train different TTS systems, since it has been proven that systems have different speaker distributions even when they are conditioned on the same speaker. The problem of discrepancies in speech features obtained from natural speech and generated by TTS models is explored in [9]. Instead of generating continuous-value features, TTS is trained to reproduce discrete representations, which are then used as input to ASR model. This approach has been shown to achieve better results in comparison to using a standard TTS which generates continuous-valued features.

In previously mentioned research outputs models are trained from scratch on publicly available datasets. However, state-of-the-art models, such as Whisper, are created using large multilingual datasets and achieve impressive results for widely spoken languages, but require additional finetuning for low-resourced languages or specific domains. An example of Whisper model finetuned using only artificial data is given in [10]. A similar approach in Whisper finetuning, with synthetic speech generated from both natural and LLM-generated text, is presented in [11].

In this paper we present further improvements to Serbian Whisper-based ASR by exploiting high-quality Serbian TTS for data augmentation. We use Serbian web-scale text corpus, SrWAC, as the input to the TTS system. Since the texts in this web-corpus are not normalized, there are many instances of numbers, abbreviations, measurement units and other ambiguous tokens. For that reason, we initially processed the input text by an LLM in order to produce texts where such tokens are expanded into orthographic words, preserving as much context as possible.

The rest of the paper is organized as follows. In Sect. 2 the used web-based text corpus, as well as its LLM-based processing, is described. The TTS system used for audio generation is presented in Sect. 3, while baseline Whisper [12]-based ASR is described in Sect. 4. The details of ASR training using TTS-generated data are presented in Sect. 5, while the experimental results are presented in Sect. 6, followed by appropriate conclusions.

2 SrWac – Description and Preprocessing

The Serbian web Corpus (srWac) was originally built by Ljubešić & Klubička in 2014 [13]. The corpus was built by scraping the Serbian top-level domain (.rs) from over 11,000 websites. Even though both Latin and Cyrillic scripts are equally used in the Serbian language, it is more common to use the Latin scripts online. All of the extracted texts in the Cyrillic script were converted to Latin, since only 16.7% of the texts were in Cyrillic. Near-duplicate texts were identified and removed, making the corpus more linguistically informative and statistically balanced. The corpus was normalized via diacritic restoration, morpho-syntactically annotated and lemmatized using the expert system described in [14]. After processing the texts scraped from the web, the corpus contains around 554 million tokens or 25 million sentences obtained from 13 million texts.

Even though texts were normalized, there remained some sentences that had to be removed because they could pose a problem if used with a Serbian TTS model, as will be discussed in more detail. Firstly, sentences that contain characters outside of the Serbian alphabet are challenging for a Serbian TTS model, since the phonetic content of the resulting synthetic speech often does correspond to the phonetic content of the same utterance in natural speech, sometimes even going beyond the limits of the Serbian phonetic inventory. For that reason, all sentences with non-ASCII characters except the Serbian letters that contain diacritics (č, ć, š, ž, đ, dž) were removed. Furthermore, it is not common for a written Serbian word to contain more than 2 identical letters in a row. Besides a relatively small number of exceptions easily identified through a dictionary, repetitions of the same letter in Serbian text are used mostly for expressive effect, emphasis, or to indicate onomatopoeia. As they were considered as potentially confusing for the system, an effort was made to exclude such sentences from the input data for the synthesizer. Other instances of potentially harmful sentences included sentences written entirely in uppercase letters, sentences containing words in which consecutive letters are separated with blank spaces, lines containing numeric-only strings, URLs and some special characters irrelevant to spoken language.

A significant portion of the corpus contained sentences in which diacritic signs were missing from letters. Namely, in informal written Serbian language diacritic signs are

often omitted from letters (“c” is written instead of “č” etc.), usually because of technical limitations such as lack of support on some keyboards. It should be noted that, as is the case with many other languages, diacritic signs in Serbian distinguish one letter from another, and the lack of a diacritic sign can alter the meaning of the word. There is, in fact, a significant number of minimal word pairs distinguished only by the presence or absence of diacritic signs. On the other hand, if diacritic signs are omitted, they are typically omitted from the entire sentence. For that reason, instead of full diacritic restoration, an alternative strategy was employed. According to [15], the total frequency of letters with diacritics in formal written Serbian is close to 3%, which implies that there is a probability of ~96% that a sentence of more than 35 characters (including spaces) without a single diacritic belongs to the informal written style in which diacritics were omitted altogether. Owing to the abundance of text material in the corpus, we could afford to exclude all such sentences from text used to generate synthetic training data for the ASR model, and dictionary-based diacritic restoration performed by the same algorithm used in the TTS front-end was carried out only on sentences with up to 35 characters without a single diacritic. Similarly, we could afford to exclude all sentences with more than 3 out-of-vocabulary words, the “vocabulary” being defined based on the morphological dictionary containing 140,000 lexemes (more than 7.8M different inflected forms) [16].

After filtering out potentially harmful sentences, each remaining sentence was post-processed, which included the capitalization of the initial letter if necessary, removing double whitespaces and punctuation normalization, i.e. conversion of excessive punctuation to a standard form (e.g. “!!!?”). Through this process, the corpus was more than halved, with 12 million sentences remaining.

A large portion of sentences contain numeric tokens and ambiguous abbreviations, which are challenging for TTS model to pronounce correctly due to their dependence on context and high degree of morphological complexity in Serbian. For that reason, we opted for a Large Language Model (LLM)-based normalizer rather than a rule heuristic. The LLM was prompted to expand numbers, Roman numerals, units and abbreviations into full orthographic words, taking the context of the sentence into account, without adding anything else to the output. The model was explicitly prompted to normalize numerical characters in all contexts, even within tokens containing mixed letters and numbers (e.g. “MP3”), and to leave acronyms intact. Ten examples of edge cases were provided to help the LLM perform better. The entire prompt is given in Appendix.

For our task we used the following models: “gpt-4o-mini”, “gemini-1.5-flash” and “deepseek-chat”. We found these models suitable for our use case since they can perform on multilingual tasks, are fast and cost effective. The Gemini and GPT models managed to expand almost all numbers, Roman numerals, units and abbreviations with relatively few exceptions, most notably mixed alphanumerics, which were later manually corrected. The DeepSeek model had less success rate and was also hallucinating new words and adding symbols that were not in the text. For that reason, almost all of the database was processed with Gemini Flash and 4o-mini.

Since lightweight models were quite effective in the said task, we also wanted to try with open source, smaller language models. The models we tested were: “Gemma3 (4B)”, “Llama3.2 (3B)”, “DeepSeek-R1 (7B)”, “phi4-mini (3.8B)” and “mistral (7B)”,

because they can perform multilingual tasks. Each model was given 5 sentences and the same prompt that was used for proprietary language models. None of the models successfully expanded all numbers, units, and abbreviations – they either failed to expand certain forms or produced incorrect outputs. Their overall performance was limited, likely due to their smaller model size. Unfortunately, we were unable to experiment with larger open-source models due to hardware constraints.

It should be noted that a small percentage of foreign words (by far mostly English ones) remained in the texts, although their pronunciation can be a difficult task for a Serbian TTS model since they do not follow relatively simple grapheme-to-phoneme conversion rules for Serbian. Although the dictionary principally contains words of the Serbian language, it was designed to handle any text in Serbian, including occasional common foreign words, most notably proper nouns such as names of persons, toponyms or brand names.

To generate the pronunciation of foreign words outside the dictionary, we utilized the G2P_lexicon Python Library, which provides pronunciations in CMU format. Since Serbian TTS models require phonemes in the Serbian phonetic system, a conversion step is implemented. Particular attention is given to the CMU phoneme AH0 (the schwa), which has no direct Serbian equivalent. A heuristic identifies adjacent phonemes and aligns them with corresponding vowels in the original input to substitute the AH0 with a more appropriate Serbian vowel. Despite these efforts, the output often contains inaccuracies, so a conversion map is dumped for manual review.

The pronunciation of acronyms in Serbian differs depending on their origin. While acronyms of English origin are spelled out according to English alphabet spelling, a vast majority of other acronyms (including those of Serbian origin) are spelled out using Serbian pronunciation of letter names. To provide variety in ASR training, half of the acronyms in input sentences are normalized using Serbian pronunciation of letter names, while the other half are normalized based on English spelling.

3 TTS and Data Generation

The Serbian text-to-speech system that was used to generate the synthetic corpus follows the canonical three-stage design of front-end, acoustic predictor and vocoder, as described in [17].

The front-end normalizes raw text, performs deterministic grapheme-to-phoneme conversion (facilitated by the language’s nearly one-to-one sound–letter correspondence), assigns prosodic tags (accents, phrase break types, sentence stress) and converts each phoneme into a high-dimensional vector of binary linguistic features that encode answers to questions such as whether the segment is a vowel, whether it carries lexical stress, or whether it precedes a phrase boundary. During model training these features are extracted from phonetically and prosodically annotated databases, while at runtime they are produced on the fly.

Acoustic prediction is handled by two deep neural networks: a duration model that estimates the length of each phoneme and an acoustic model that, conditioned on the linguistic features and the predicted durations, outputs vocoder parameters. Both networks consist of three feed-forward layers with ReLU activation followed by a single

LSTM layer; they are trained in a multi-speaker configuration and subsequently fine-tuned to individual voices, an approach that shortens speaker adaptation time compared with training from scratch.

The original implementation used the deterministic WORLD vocoder, but it has been superseded by a HiFi-GAN neural vocoder [18]. The model consists of a fully convolutional generator that upsamples 80-channel mel-spectrograms via transposed convolutions, together with two complementary discriminator groups. A multi-period discriminator evaluates slices of the signal taken at different hop intervals, making it sensitive to pitch-related periodicity, whereas a multi-scale discriminator inspects the waveform at several temporal resolutions to enforce local spectral detail. For Serbian TTS the universal HiFi-GAN, originally trained on English, is further fine-tuned on spectrograms emitted by the acoustic predictor rather than on natural speech directly; this guided adaptation aligns the generator with the statistical characteristics of the upstream network and yields noticeably higher naturalness, as previously demonstrated for Serbian TTS in [17].

The combination of a linguistically rich front-end, a two-stage acoustic predictor and a speaker-adapted HiFi-GAN vocoder constitutes the highest-quality and most widely deployed multispeaker Serbian TTS to date and provides the natural-sounding synthetic speech leveraged in the present ASR fine-tuning experiments. This system was used to produce the synthetic corpus of Serbian, using 14 distinct voices, 9 female and 5 male. For each sentence selected from the cleaned srWaC text pool, the generation script sampled three parameters stochastically:

- a voice identity drawn uniformly from the fourteen speakers,
- a speaking-rate multiplier chosen from a continuous range of 0.7 to 1.3 (approximately $\pm 30\%$ around the default rate), and
- a fundamental frequency offset applied linearly within $\pm 30\%$ of the speaker’s baseline f_0 .

The permissible f_0 range for each voice was verified by informal listening to ensure that shifts did not introduce robotic artefacts or compromise intelligibility. This randomized parameterization yields a speech collection whose variation in timbre, tempo and intonation approximates that of spontaneous conversational speech, while preserving naturalness throughout. Every synthesized utterance was saved as a 16-kHz, 16-bit mono WAV file and paired with its reference text in Whisper-compatible JSON manifest format, enabling seamless integration into the subsequent ASR fine-tuning pipeline. The final TTS-generated corpus comprises slightly more than one million audio files, corresponding to over 1500 h of speech.

4 Whisper-Based Serbian ASR

Our experiments start from the Whisper-medium configuration, a 769 million-parameter encoder-decoder transformer that belongs to the Whisper family of end-to-end automatic-speech-recognition (ASR) and speech-translation models. Whisper was originally trained on approximately 680,000 h of multilingual audio-text pairs covering 97 languages, with more than 400,000 h in English and only 28 h in Serbian plus 91 h

in Croatian (a mutually intelligible South Slavic language originating from the same pluricentric Serbo-Croatian base). This severe imbalance is reflected in the model’s error profile: performance in English is state-of-the-art, whereas recognition accuracy for Serbian remains well below that level (on CV16 WER is 85.6%, while on FLEURS WER is 44.9%).

Whisper’s training pipeline constrains every audio clip to ≤ 30 s sampled at 16 kHz. Shorter clips are zero-padded, and inference is likewise performed in 30-s windows. While this strategy yields a streamlined model, it introduces boundary effects when longer recordings must be processed in contiguous segments. Although Whisper can output timestamp predictions, published work has shown that these offsets are often imprecise, motivating several alignment-oriented extensions [19].

Architecturally, Whisper employs a log-Mel spectrogram front end, followed by a transformer encoder whose hidden sequence is consumed by a transformer decoder that generates text tokens in an autoregressive fashion. The token inventory, shared across all 97 languages, originates from the ChatGPT vocabulary [20]. For Serbian this design is double-edged: on the one hand, the model benefits from a very large joint lexicon; on the other, a single token may correspond to an unpredictable cluster of one to three phonemes, which occasionally results in out-of-vocabulary word forms.

Whisper is publicly released in five size tiers, ranging from tiny (39 M parameters) to large (1.55 B). We selected the medium tier because it offers a favourable trade-off between robustness and computing cost; a single consumer-grade GPU with approximately 10 GB of memory is sufficient for both inference and further fine-tuning.

To obtain a competitive Serbian baseline we fine-tuned Whisper-medium on around 1300 h of manually transcribed audio supplied by the company AlfaNum from Novi Sad, Serbia. The corpus combines audiobook narration and radio/television content in Serbian, and includes a smaller section in Croatian. Average segment duration is five seconds, and all files are sampled at 16 kHz, matching Whisper’s default front-end. This adaptation step narrows the performance gap caused by the original data imbalance but still leaves systematic errors on low-frequency vocabulary and atypical prosody, which are issues that we address by augmenting the decoder with diversified synthetic speech.

5 Integrating Synthetic Speech into the Whisper-Based Serbian ASR

The experiments start from a Whisper-medium model that had already been fine-tuned on roughly 1300 h of manually transcribed Serbian and Croatian speech collected from radio programs, television shows, and audiobooks. Although this baseline performs well, detailed error analysis revealed persistent mistakes on low-frequency words. To mitigate these errors we added the synthetic corpus described in the preceding chapters, stored in Whisper’s 16 kHz manifest format so that the original training pipeline required no substantive changes.

A key design decision was to freeze the entire encoder (every convolutional block and transformer layer) while updating only the decoder and the output projection. Synthetic speech, although perceptually natural, still differs from recorded speech in spectral detail and micro-prosody. If the encoder were updated on this material, its acoustic

embeddings could shift toward features specific to the TTS engine, weakening its ability to handle background noise, reverberation, and channel variations present in real recordings. Freezing the encoder therefore preserves these representations and confines adaptation to the language-model component, where expanded lexical and intonational coverage is required. All available Serbian TTS voices were utilized in order to capture differences in pronunciation and speech dynamics among speakers, which could potentially influence the decoder. Since the encoder was kept frozen, the impact of voice timbre and the imbalance between male and female voices is negligible.

With the encoder frozen, the decoder was fine-tuned on a combined dataset comprising the original 1300 h of real speech data and the 1500 h of synthetic speech. Fine-tuning proceeded for 3 epochs on a mixture of the full synthetic set and the original real-speech data, but gave the best result after the first epoch. Training used AdamW [21] optimization with a 500-step warm-up and a learning rate of 5×10^{-6} with cosine scheduler type and 0.1 weight decay. The batch size was 8 with 2 gradient accumulation steps, i.e. effective batch of 16. The decoder converged rapidly, acquiring the intended vocabulary and prosodic cues. Crucially, validation on natural recordings showed no evidence of the acoustic drift often observed when encoders are allowed to adapt to large quantities of synthetic speech.

6 Experiments

To quantify the impact of the synthetic data, we evaluated both the baseline and the augmented recognizer on public benchmarks and on internal, real-world recordings. The public portion comprised the Serbian test subset of Common Voice v16 (1543 sentences), which contains crowd-sourced read speech from speakers with a wide range of regional accents, and the Serbian test partition of FLEURS, an n-way parallel corpus derived from the FLoRes-101 translation benchmark that offers studio-quality recordings of 700 sentences read by volunteer speakers.

For application-level validation we collected three domains that mirror everyday production traffic: multi-speaker meetings (corporate briefings, doctor-patient interactions, and courtroom exchanges), broadcast programs (television series and talk shows), and short field reports plus customer inquiries recorded in call-center IVR scenarios. The evaluation set contains 25 recordings with a total duration of ~50 min in meetings (5–23 min per recording), ~127 min in broadcasts (20–58 min per recording), and ~4 min in reports/inquiries (3–44 s per recording); the first two domains comprise long sessions, whereas the last category consists of utterances among which some are only a few seconds in length. Word-error rate (WER) was computed after standard text normalization, which removed capitalization and punctuation.

The model fine-tuned with synthetic speech achieved lower WER in every domain except broadcasts (Table 1). On Common Voice the error rate fell from 7.1% to 6.8%, whereas FLEURS remained unchanged at 10.6%. Meeting recordings improved from 22.9% to 19.6%, and reports plus call-center inquiries from 8.7% to 4.3%. Broadcast speech showed a small, statistically non-significant decrease from 12.6% to 12.2%. Crucially, no domain displayed a meaningful deterioration, indicating that freezing the encoder successfully guards against over-fitting to synthetic artefacts. Decoder-only fine-tuning had no measurable effect on throughput or latency: inference speed remained

identical to the baseline. Whisper-based recognizers are intended primarily for offline or batch processing rather than low-latency applications, their transcription speed is still highly competitive: on recent GPUs they run roughly 10–20× faster than real time when deployed with efficient inference libraries such as `whisper.cpp` or `FasterWhisper`.

Table 1. WER and CER comparison on different test sets and domains.

	Whisper-based Serbian ASR – baseline		Whisper-based Serbian ASR – finetuned to TTS data	
	WER (%)	CER (%)	WER (%)	CER (%)
CV16	7.1	2.9	6.8	2.7
FLEURS	10.6	6.0	10.6	6.0
Meetings	22.9	9.6	19.6	9.1
Broadcast	12.2	7.9	12.6	7.6
Reports/inquiries	8.7	2.9	4.3	1.6

7 Conclusion

In this paper Serbian Whisper-based ASR system exploiting TTS-generated data is introduced. The input sentences for TTS are extracted from publicly available web-based TTS corpus, which is precisely curated, to exclude any potentially problematic sentences for TTS, and additionally processed by an LLM, to overcome some specifics of the Serbian language in process of text-normalization. The baseline in experiments was a medium-sized Serbian Whisper-ASR model finetuned from original OpenAI’s medium model on 1300 h of Serbian speech. This baseline model was further finetuned by 1500 h of speech synthesized in 14 distinct voices. The model finetuned on synthetic speech significantly surpasses the performance of the baseline model on most test sets, confirming data augmentation by synthetic speech to be a valid approach to improve the performance of automatic speech recognition in under-resourced languages.

Acknowledgments. This research was supported by the Science Fund of the Republic of Serbia, Grant No. 7449, Multimodal multilingual human-machine speech communication, AI-SPEAK.

Appendix: Prompt Used to Instruct the Language Model

The prompt is given in it’s original form in Serbian as well is in English for reproducibility.

“U rečenicama koje ću navoditi u narednim promptovima ekspanđuj broj, jedinicu mere i skraćenice u pune reči, uzimajući u obzir kontekst. Nemoj dodavati dodatne komentare, samo konvertuj rečenicu koju unesem. Za brojeve ne postoje izuzeci, uvek ih sve konvertovati u reči. Ukoliko broj stoji uz slovo, pretvori ga u reč (B92 u B devedeset

dva). Vodi računa o padežima. Akronime nemoj obavezno konvertovati. Samo ako si potpuno siguran da u datom kontekstu imaju određeno značenje. Rimske brojeve takođe konvertovati u reči.

Primeri:

Prešao je 57 km za 3 h.

Prešao je pedeset sedam kilometara za tri sata.

On je doc. na pravnom fakultetu već 10 g.

On je docent na pravnom fakultetu već deset godina.

Na 39. km od polaska mu se desila nesreća.

Na trideset devetom kilometru od polaska mu se desila nesreća.

G. Kostić, 39-godišnji van. prof. održaće predavanje.

Gospodin Kostić, trideset devetogodišnji vanredni profesor održaće predavanje.

EU je osnovana 1. novembra 1993. g.

Evropska Unija je osnovana prvog novembra hiljadu devetsto devedeset treće godine.

Luj XIV je preminuo mnogo pre I svetskog rata.

Luj četrnasti je preminuo mnogo pre prvog svetskog rata.

Lambdacizam predstavlja nepravilan izgovor glasa L.

Lambdacizam predstavlja nepravilan izgovor glasa L.

U 2017. godini sam upisao FTN.

U dve hiljade sedamnaestoj godini sam upisao FTN.

F1 se emituje od 1950. do 2024.

F jedan se emituje od hiljadu devetsto pedesete do dve hiljade dvadeset četvrte.

Izmerio je 29 mmHg.

Izmerio je dvadeset devet milimetara živinog stuba.”

“In the sentences that I will provide in the following prompts, expand the number, unit of measure, and abbreviations into full words, taking the context into account. Do not add any additional comments, just convert the sentence I send. For numbers, there are no exceptions, always convert them fully into words. If a number is next to a letter, convert it into words (e.g., B92 becomes B ninety two). Pay attention to grammatical cases. Acronyms should not necessarily be converted, only if you are completely certain they have a specific meaning in the given context. Roman numerals should also be converted into words.

Examples:

He covered 57 km in 3 h.

He covered fifty-seven kilometers in three hours.

He has been an asst. prof. at the Faculty of Law for 10 y.

He has been an assistant professor at the Faculty of Law for ten years.

An accident happened to him at 39. km from the start.

An accident happened to him at the thirty-ninth kilometer from the start.

Mr. Kostić, a 39-year-old assoc. prof. will give a lecture.

Mister Kostić, a thirty-nine-year-old associate professor, will give a lecture.

EU was founded on 1. November 1993.

The European Union was founded on the first of November, nineteen ninety-three.

Louis XIV died long before WWI.

Louis the Fourteenth died long before the First World War.

Lambdacism refers to the incorrect pronunciation of the sound L.

Lambdacism refers to the incorrect pronunciation of the sound L.

In 2017. I enrolled at FTN.

In the year two thousand seventeen, I enrolled at FTN.

F1 has been broadcast from 1950 to 2024.

F one has been broadcast from nineteen fifty to two thousand twenty-four.

He measured 29 mmHg.

He measured twenty-nine millimeters of mercury.

References

1. Prabhavalkar, R., Hori, T., Sainath, T.N., Schlüter, R., Watanabe, S.: End-to-end speech recognition: a survey. *IEEE/ACM Trans. Audio Speech Lang. Process.* **32**, 325–351 (2023)
2. Mimura, M., Ueno, S., Inaguma, H., Sakai, S., Kawahara, T.: Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 477–484. IEEE (2018)
3. Shen, J., et al.: Natural TTS synthesis by conditioning wavenet on MEL spectrogram predictions. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4779–4783. IEEE (2018)
4. Zheng, X., Liu, Y., Gunceler, D., Willett, D.: Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end ASR systems. In: 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ICASSP 2021, pp. 5674–5678. IEEE (2021)

5. Aich, A.: Elastic weight consolidation (EWC): nuts and bolts. arXiv preprint [arXiv:2105.04093](https://arxiv.org/abs/2105.04093) (2021)
6. Fazel, A., et al.: Synthesr: unlocking synthetic data for speech recognition. arXiv preprint [arXiv:2106.07803](https://arxiv.org/abs/2106.07803) (2021)
7. Ueno, S., Mimura, M., Sakai, S., Kawahara, T.: Multi-speaker sequence-to-sequence speech synthesis for data augmentation in acoustic-to-word speech recognition. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), ICASSP 2019, pp. 6161–6165. IEEE (2019)
8. Yuen, K.C., Li, H., Siong, C.E.: ASR model adaptation for rare words using synthetic data generated by multiple text-to-speech systems. In: 2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), pp. 1771–1778. IEEE (2023)
9. Ueno, S., Mimura, M., Sakai, S., Kawahara, T.: Data augmentation for ASR using TTS via a discrete representation. In: 2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 68–75. IEEE (2021)
10. Vásquez-Correa, J.C., Arzelus, H., Martín-Doñas, J.M., Arellano, J., Gonzalez-Docasal, A., Álvarez, A.: When whisper meets TTS: domain adaptation using only synthetic speech data. In: International Conference on Text, Speech, and Dialogue, pp. 226–238. Springer, Cham (2023)
11. Cornell, S., Darefsky, J., Duan, Z., Watanabe, S.: Generating data with text-to-speech and large-language models for conversational speech recognition. arXiv preprint [arXiv:2408.09215](https://arxiv.org/abs/2408.09215) (2024)
12. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning, pp. 28492–28518. PMLR (2023)
13. Ljubešić, N., Klubička, F.: {BS, HR, SR} WAC-web corpora of Bosnian, Croatian and Serbian. In: Proceedings of the 9th Web as Corpus Workshop (WaC-9), pp. 29–35 (2014)
14. Vlado, D., Milan, S., Nikša, J., Marko, J., Radovan, O., Darko, P.: Speech technologies for Serbian and Kindred South Slavic languages. In: Shabtai, N.R. (ed.) Advances in Speech Recognition, SCIYO, pp. 141–164 (2010). <https://doi.org/10.5772/10115>. ISBN 978-953-307-097-1
15. Babić, V.: The frequency of lowercase and uppercase letters, bigrams, and trigrams in the Serbian language. InfoM, no. 79–80/2024, pp. 22–26 (2024)
16. Sečujski, M.: Accentuation dictionary of Serbian intended for text-to-speech synthesis. In: Proceedings of the Digital Image and Signal Processing Conference on DOGS 2002, Bečej, Serbia, pp. 17–20 (2002). (in Serbian)
17. Suzić, S., Pekar, D., Sečujski, M., Nosek, T., Delić, V.: HiFi-GAN based text-to-speech synthesis in Serbian. In: 2022 30th European Signal Processing Conference (EUSIPCO), pp. 2231–2235. IEEE (2022)
18. Kong, J., Kim, J., Bae, J.: HiFi-GAN: generative adversarial networks for efficient and high fidelity speech synthesis. Adv. Neural. Inf. Process. Syst. **33**, 17022–17033 (2020)
19. Wagner, L., Thallinger, B., Zusag, M.: CrisperWhisper: accurate timestamps on verbatim speech transcriptions. arXiv preprint, [arXiv:2408.16589](https://arxiv.org/abs/2408.16589) (2024)
20. Wu, T., et al.: A brief overview of ChatGPT: the history, status quo and potential future development. IEEE/CAA J. Automatica Sinica **10**(5), 1122–1136 (2023)
21. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint [arXiv:1711.05101](https://arxiv.org/abs/1711.05101) (2017)



Domain Knowledge and Language Embeddings for Low-Resource Multilingual Phoneme ASR

Anton Legchenko^(✉) and Ivan Bondarenko

Novosibirsk State University, Novosibirsk, Russia
{a.legchenko,i.bondarenko}@g.nsu.ru

Abstract. This paper presents methods to improve the accuracy and robustness of multilingual automatic speech recognition (ASR) systems transcribing speech into International Phonetic Alphabet (IPA) sequences. The development of such systems faces considerable challenges, including linguistic diversity, pronunciation variability, and especially the scarcity of high-quality annotated resources for many languages, which hinders model generalization to unseen linguistic domains. We propose a framework that explicitly integrates prior linguistic knowledge into the model training process and leverages auxiliary information via hierarchical multi-task learning (HMTL). The method decomposes phoneme recognition into several levels of abstraction, thus enabling the model to capture both language-independent and language-specific phonetic patterns. Furthermore, we introduce and compare two types of language vector representations, obtained respectively from acoustic signals and from phonetic transcriptions, and evaluate their utility as auxiliary inputs, particularly for low-resource and zero-shot scenarios. Experiments were conducted on multilingual corpora with both high- and low-resource languages, employing a pre-trained Wav2Vec 2.0 architecture as the base model. Baseline models were fine-tuned using Connectionist Temporal Classification (CTC) loss without auxiliary information. Phoneme Error Rate (PER) was used for evaluation in both in-domain and out-of-domain settings. The results demonstrate a relative improvement in recognition accuracy of 7–10% for most scenarios, and an improvement exceeding 20% for out-of-domain languages under reduced training data conditions.

Keywords: Speech recognition · Multilingual ASR · Phoneme recognition · Language embeddings · Hierarchical multi-task learning · IPA transcription

1 Introduction

The automatic recognition of speech in multiple languages and transcription into the International Phonetic Alphabet (IPA) is fundamental for a range of appli-

cations, including phonetic research, language documentation, language learning, and assistive speech technologies. Despite recent progress in deep learning-based ASR, achieving robust and accurate IPA transcription across linguistically diverse and low-resource languages remains a formidable challenge [4–6]. Primary obstacles include the vast diversity of phonetic inventories, significant pronunciation variability driven by dialect, accent, and speaker idiosyncrasies, and the limited availability of annotated corpora for most languages. These factors impede both the generalization ability of models and their applicability to previously unseen languages.

State-of-the-art multilingual ASR systems are often based on large-scale architectures such as Wav2Vec 2.0 [3] and Whisper [22], which utilize massive amounts of unlabeled or weakly labeled data and transformer-based models. While these models achieve high performance on languages well-represented in training data, their performance on low-resource languages is considerably lower, with a tendency to overfit to the available data and limited capability to generalize phonetic patterns across languages. Inconsistencies in phonetic annotation practices and the inherent noise in multilingual speech corpora further complicate training and evaluation [30].

To address these limitations, this work introduces a unified framework that incorporates domain-specific prior linguistic knowledge and auxiliary language representations into ASR models. Specifically, we propose a hierarchical multi-task learning (HMTL) approach [24] that structures phoneme recognition as a sequence of prediction tasks at varying levels of abstraction. The model simultaneously learns to predict phoneme classes (based on IPA phonetic groupings) and the target phoneme sequences, promoting the learning of both universal and language-specific phonetic representations. In addition, we investigate two types of language embedding vectors, derived either from raw speech or from phonetic transcriptions, and employ them as auxiliary inputs to facilitate model adaptation, particularly for languages with little or no training data.

We systematically evaluate these methods using the Common Voice corpus [2] with a diverse selection of language pairs and training regimes. Experimental results demonstrate that both the integration of linguistic structure via HMTL and the use of language embeddings lead to substantial reductions in Phoneme Error Rate (PER), especially in zero-shot and few-shot settings. Notably, the combination of both approaches results in the largest gains, indicating their complementary nature.

The remainder of the paper is organized as follows. Section 2 reviews related work on multilingual phoneme recognition, hierarchical learning, and language embedding techniques for ASR. Section 3 details our methodology, including model architectures, embedding extraction, and training strategies. Section 4 presents experimental settings and results. Finally, Sect. 5 summarizes the findings and outlines future research directions.

2 Related Work

The development of multilingual ASR systems, particularly those capable of generating phonetic (IPA) transcriptions, has been an area of intensive research over recent decades. Initial approaches predominantly relied on statistical methods such as Hidden Markov Models (HMMs) for both acoustic and language modeling [21]. The advent of deep learning, however, has markedly advanced the field, enabling substantial gains in both accuracy and generalizability.

Deep neural networks, including recurrent (RNN) [10, 17] and convolutional (CNN) [7] architectures, have proven effective at capturing the complex temporal and spectral dynamics inherent to speech. More recently, transformer-based models [8, 29], such as Wav2Vec 2.0 [3] and Whisper [22], have emerged as state-of-the-art in multilingual speech recognition, utilizing large-scale self-supervised pretraining to learn rich, universal acoustic representations.

Several studies have addressed the challenge of generating IPA transcriptions by employing grapheme-to-phoneme (G2P) conversion tools, including eSpeaking [1] and Phonetisaurus [19], which facilitate the construction of large multilingual datasets [30]. Nevertheless, the effectiveness of such tools can vary across languages, and inconsistencies in transcription standards can introduce noise and reduce the overall quality of training data.

Multi-task learning (MTL) has gained attention as an effective strategy for leveraging auxiliary tasks to enhance ASR models [25, 32]. In particular, hierarchical multi-task learning (HMTL) has shown promise in modeling linguistic structure at multiple abstraction levels [23, 24], enabling models to simultaneously capture language-agnostic and language-specific phonetic features. The use of auxiliary information, such as language identity or embeddings, has also been explored, with several works proposing the integration of language vectors learned from either raw speech [15, 28] or phonetic transcriptions [14, 18] as conditioning inputs for ASR or TTS models.

Previous multilingual phoneme recognition approaches have explicitly incorporated universal phonetic knowledge through fixed mappings or hard constraints. For example, Li et al. [13] propose a multilingual allophone system for universal phone recognition that relies on a manually defined mapping between each language’s phonemes and a set of language-independent phonetic units. Similarly, Yen et al. [31] exploit universal articulatory features (such as manner and place of articulation) by constructing deterministic attribute-to-phoneme mapping matrices that impose strict, rule-based constraints on the output phoneme predictions. Both methods leverage predefined phoneme-to-feature correspondences to guide multilingual models.

In contrast, our approach does not require any fixed phoneme-to-attribute mapping or external phonetic inventory. Instead, we integrate phonetic knowledge through an auxiliary IPA-class prediction task inserted at an intermediate Transformer layer, rather than enforcing constraints at the output layer. This provides a soft, learned form of articulatory guidance within a fully end-to-end model, incorporating the phonetic hierarchy in a data-driven manner without explicit rules or external structures. Furthermore, our experiments show

that placing the auxiliary task at an earlier layer of the network yields better generalization to unseen (out-of-domain) languages compared to applying such supervision only at the output, highlighting the benefit of early-layer phonetic supervision in multilingual phoneme recognition.

3 Methodology

To address the outlined challenges in multilingual phoneme recognition, we propose and systematically compare several approaches, focusing on the integration of domain-specific linguistic knowledge and auxiliary language representations into the ASR pipeline. All methods are benchmarked against a strong baseline: the Wav2Vec 2.0 XLSR-53 architecture [3], which has demonstrated state-of-the-art results for multilingual speech processing [20, 30].

3.1 Baseline Model and Evaluation Protocol

The baseline system utilizes a pre-trained Wav2Vec 2.0 XLSR-53 encoder, which is fine-tuned for phoneme sequence prediction on a multilingual corpus transcribed in IPA. The training objective is the Connectionist Temporal Classification (CTC) loss [9], suitable for sequence alignment tasks where the correspondence between input frames and output labels is unknown. The CTC loss is formally defined as:

$$\mathcal{L}_{\text{CTC}} = -\log P(\mathbf{y} \mid \mathbf{x}) = -\log \sum_{\mathbf{a} \in \mathcal{A}(\mathbf{x}, \mathbf{y})} P(\mathbf{a} \mid \mathbf{x}), \quad (1)$$

where \mathbf{x} is the input audio, \mathbf{y} is the reference phoneme sequence, and $\mathcal{A}(\mathbf{x}, \mathbf{y})$ denotes the set of all possible alignments.

For objective evaluation, we employ the Phoneme Error Rate (PER), calculated as the normalized Levenshtein distance between predicted and reference phoneme strings:

$$\text{PER} = \frac{S + D + I}{N} \times 100\%, \quad (2)$$

where S is the number of substitutions, D is deletions, I is insertions, and N is the total number of reference phonemes.

3.2 Hierarchical Multi-task Learning for Phoneme Recognition

In order to exploit universal phonetic structures, we introduce a hierarchical multi-task learning (HMTL) framework. This architecture augments the main phoneme prediction task with an auxiliary objective: predicting higher-level phoneme classes derived from the IPA taxonomy (such as rows or columns of IPA tables for consonants and vowels). Each phoneme in the target sequence is replaced by its respective class label to construct the auxiliary sequence. An

additional classification head, attached at an intermediate layer, is optimized jointly with the primary output using a combined loss:

$$\mathcal{L}_{\text{HMTL}} = \mathcal{L}_{\text{CTC}}^{\text{phoneme}} + \lambda \cdot \mathcal{L}_{\text{CTC}}^{\text{class}}, \quad (3)$$

where λ controls the auxiliary loss weight.

This hierarchical decomposition encourages the model to learn shared phonetic abstractions across languages, facilitating better generalization—especially in the presence of limited or noisy data. The architectural scheme is depicted in Fig. 1.

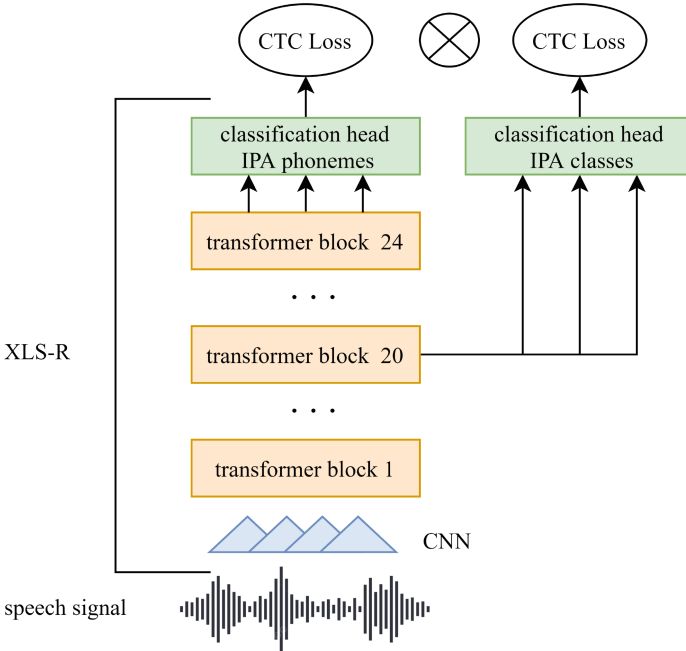


Fig. 1. Schematic overview of the hierarchical multi-task model for multilingual phoneme recognition.

Effect of Auxiliary-Head Placement. To identify the most informative level for the auxiliary prediction we varied (i) the layer at which the second CTC head is attached and (ii) whether IPA *rows* or *columns* are used as class labels. Table 1 shows that attaching the auxiliary head to the **20-th transformer layer** and using *column* groupings for both consonants and vowels gives the lowest PER on held-out languages.

Table 1. PER metrics on test set for 12 training languages, depending on HMTL configuration.

Layer of additional heads/IPA consonants/IPA vowels	PER out of domain		PER in domain	
	ES	CV	ES	CV
Baseline	0.358	0.348	0.113	0.117
16/Columns/Columns	0.331	0.271	0.107	0.115
20/Columns/Columns	0.323	0.264	0.103	0.109
24/Columns/Columns	0.345	0.294	0.111	0.116
16/Rows/Rows	0.339	0.279	0.111	0.115
20/Rows/Rows	0.334	0.270	0.108	0.112
24/Rows/Rows	0.341	0.292	0.109	0.116
20/Columns/Rows	0.329	0.267	0.105	0.110
20/Rows/Columns	0.337	0.282	0.110	0.115

3.3 Language Embedding Extraction

To further enhance cross-lingual adaptation and facilitate few-shot or zero-shot recognition, we incorporate language embedding vectors as auxiliary inputs. Two complementary approaches for embedding extraction are explored:

Speech-Based Embeddings. We utilize the Wav2Vec 2.0 XLSR-53 model to derive utterance-level representations using an XVector-style head [27]. The model is optimized using the triplet loss [11], which encourages embeddings of utterances from the same language to be close, and those from different languages to be distant. Triplets are constructed by selecting anchor, positive (same language), and negative (different language) samples:

$$\mathcal{L}_{\text{triplet}} = \max(0, d(\mathbf{e}_{\text{anchor}}, \mathbf{e}_{\text{positive}}) - d(\mathbf{e}_{\text{anchor}}, \mathbf{e}_{\text{negative}}) + \alpha), \quad (4)$$

where $d(\cdot, \cdot)$ is a distance metric (e.g., cosine), and α is the margin.

Transcription-Based Embeddings. We obtain sequence-level language representations from IPA phonetic transcriptions using PhoneBERT [14], a BERT-style model pre-trained with masked language modeling on multilingual phoneme sequences. The embedding for a transcription is computed by mean pooling over the output token representations. The overall workflow for extracting and incorporating language embeddings—whether from raw audio or from phonetic transcriptions—into the ASR model is illustrated in Fig. 2.

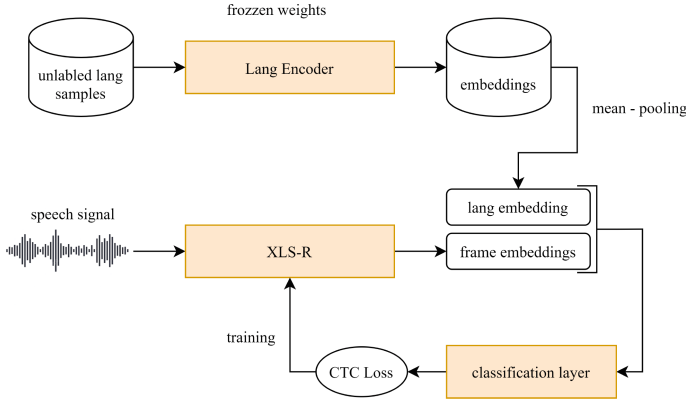


Fig. 2. Inference and training scheme with language auxiliary information based on available language samples.

3.4 Integration of Auxiliary Language Information

During ASR model training and inference, language embeddings (either from speech or from transcriptions) are concatenated or otherwise fused with the main acoustic representations, providing explicit context regarding the target language. Both instance-level (utterance-specific) and averaged (language-level) embeddings are evaluated. Additional control experiments utilize one-hot language labels to isolate the contribution of learned embedding representations.

3.5 Experimental Setup

All experiments are conducted on subsets of the Mozilla Common Voice corpus [2], with careful curation to balance the amount of training data per language and to allow held-out language evaluation. G2P conversion to IPA is performed using Espeak-ng [1]. Optimization is performed using AdamW [12] with cyclical learning rate scheduling [26], and for model selection and stopping criteria, we employed two validation strategies: A conventional early stopping method using a development set from the same language as the evaluation (in-language validation), and a cross-language validation strategy where languages are treated as folds. In the CV setup, the model’s stopping point is determined by validation performance on held-out language data (i.e., on languages not seen in training), encouraging better generalization to unseen languages. We report results for both strategies in our tables, with ‘ES’ and ‘CV’ denoting the respective approach.

All models were implemented using the PyTorch framework and trained on single NVIDIA Tesla A100 80 GB GPU.

4 Experimental Results

4.1 Evaluation Metrics

The main evaluation criterion is the Phoneme Error Rate (PER), computed as the normalized Levenshtein distance between predicted and reference IPA sequences (see Sect. 3). Lower PER values indicate better phoneme recognition accuracy.

4.2 Comparison of Methods

We report results for the following model variants:

- **Baseline:** Wav2Vec 2.0 XLSR-53 fine-tuned with CTC loss, without auxiliary inputs.
- **Hierarchical Multi-Task Learning (HMTL):** Augmented with an auxiliary CTC objective for IPA-based phoneme classes, as detailed in Sect. 3.
- **Speech-based language embeddings:** Integration of averaged language vectors derived from speech.
- **Transcription-based language embeddings:** Integration of language vectors obtained from IPA transcriptions.
- **Combined approaches:** Models employing both HMTL and auxiliary embeddings.

Hierarchical Multi-task Learning. Table 2 summarizes the impact of HMTL across different training set sizes. The hierarchical approach consistently outperformed the baseline in both in-domain and out-of-domain settings, with particularly strong improvements on unseen languages (PER reduction up to 20% in low-resource conditions).

Table 2. Phoneme Error Rate (PER) for baseline and HMTL models across various training set sizes.

Languages	Approach	PER Out-of-Domain	PER In-Domain
4	Baseline	0.447	0.151
4	HMTL	0.396	0.142
8	Baseline	0.386	0.132
8	HMTL	0.316	0.111
12	Baseline	0.358	0.113
12	HMTL	0.323	0.103

Auxiliary Language Embeddings. Both speech-based and transcription-based language embeddings improved recognition performance, with averaged embeddings showing the best generalization to unseen languages. A further ablation using one-hot language identifiers confirmed that the benefit stems from the structured embedding space, not simply language label awareness. Results are summarized in Table 3.

Table 3. PER for models using speech and transcription-based language embeddings (for 4, 8, 12 training languages).

Languages	Model	PER Out-of-Domain	PER In-Domain
4	Baseline	0.447	0.151
4	Speech Embedding	0.421	0.147
4	Transcr. Embedding	0.431	0.142
8	Baseline	0.386	0.132
8	Speech Embedding	0.357	0.127
8	Transcr. Embedding	0.361	0.125
12	Baseline	0.358	0.113
12	Speech Embedding	0.321	0.104
12	Transcr. Embedding	0.321	0.103

Combined Approaches and Detailed Evaluation. Combining HMTL and language embeddings yielded the lowest PER, both in-domain and for held-out languages. Detailed evaluation on four held-out languages is shown in Table 4. Notably, the largest relative improvements were observed for out-of-domain languages under reduced training data, reaching more than 20% reduction compared to the baseline.

Scaling to 26 Languages. Finally, we repeated the full experimental pipeline on a 26-language training set (Common Voice languages with down sampling to ≤ 10 h of data). The combined approach achieved an **in-domain PER of 0.088** and an average **out-of-domain PER of 0.251**, representing a $\approx 7\%$ relative improvement over the baseline (0.093/0.265). This shows that the proposed techniques remain effective even when the language inventory is doubled.

4.3 Analysis of Language Embedding Substitution

To examine the linguistic structure encoded by the learned language vectors, additional experiments substituted embeddings of held-out languages with those of related or unrelated languages. Results demonstrated that using embeddings from closely related languages (e.g., substituting German with Swedish or

Table 4. Best PER on held-out languages: French – fr, German – de, Catalan – ca, Japanese – ja.

Approach	PER in-domain	PER fr	PER de	PER ca	PER ja
12 Training Languages					
Baseline	0.113	0.348	0.314	0.361	0.423
HMTL	0.103	0.264	0.253	0.323	0.392
Speech embeddings	0.106	0.301	0.281	0.333	0.391
Transcription embeddings	0.103	0.321	0.279	0.337	0.385
Speech embeddings + transcription embeddings	0.098	0.311	0.275	0.324	0.373
Transcription embeddings + HMTL	0.092	0.261	0.271	0.339	0.378
Speech embeddings + HMTL	0.091	0.252	0.276	0.341	0.372
Combination of all methods	0.091	0.249	0.245	0.325	0.361
26 Training Languages					
Baseline	0.093	0.265	0.249	0.311	0.323
HMTL	0.089	0.249	0.238	0.301	0.315
Combination of all methods	0.088	0.251	0.234	0.292	0.281

English) led to moderate performance degradation, while unrelated languages (e.g., Japanese or Russian) resulted in significant loss of accuracy. This finding underscores the capacity of the embedding space to encode genealogical and phonetic relationships.

4.4 Visualization and Embedding Evaluation

Language embedding spaces were visualized using the UMAP technique [16], confirming that phonetically or genealogically related languages cluster closely. Quantitative evaluation with logistic regression classifiers on embedding vectors yielded language-identification accuracies of **0.88** for the speech-based embeddings and **0.917** for the transcription-based embeddings, confirming the high discriminative power of both representations.

Embedding Substitution Experiment. The result of the test data for the speech-based embedding model is shown in Fig. 3. To probe the linguistic structure captured by the vectors, we replaced the German embedding at test time with vectors from typologically related and unrelated languages. Table 5 confirms that genealogical proximity matters: Swedish (same Germanic family) even *improved* performance, whereas distant languages seriously degraded it.

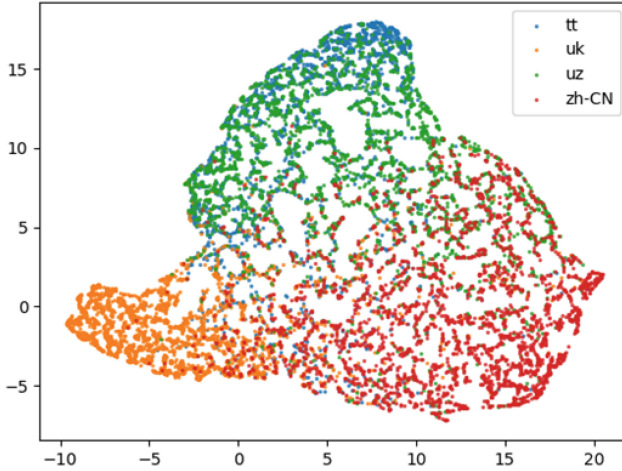


Fig. 3. Visualization of the two-dimensional UMAP projection of test data embeddings.

Table 5. PER on German test utterances when the German embedding is replaced by another language vector.

Substituted embedding	PER on de
Native German vector	0.234
Swedish (sv-SE)	0.225
English (en)	0.243
Russian (ru)	0.295
Japanese (ja)	0.301

4.5 Computational Resources

Experiments required approximately one hour per training epoch on a single NVIDIA A100 GPU for 12 languages, with computational overhead increasing by up to 20% when auxiliary losses were employed. Most models converged within 1–3 epochs depending on the validation strategy.

5 Conclusion

This study has addressed the problem of enhancing multilingual automatic speech recognition (ASR) systems for phoneme-level transcription in the International Phonetic Alphabet (IPA), with special attention to linguistic diversity and the challenge of low-resource languages. We proposed an integrated framework combining hierarchical multi-task learning (HMTL), which incorporates domain-specific phonetic abstractions, with auxiliary language representations extracted from both speech and phonetic transcriptions.

Experimental evaluation on the Common Voice corpus demonstrated that each proposed component contributes to improved generalization, yielding consistent reductions in phoneme error rate (PER) on both in-domain and out-of-domain data. In particular, the combined approach produced a relative improvement of 7–10% in recognition accuracy across most test scenarios, and more than 20% for held-out languages under low-resource conditions. Additional analysis revealed that language embeddings effectively capture genealogical and phonetic relations, facilitating zero-shot transfer to typologically similar languages.

Future work will explore more sophisticated integration of linguistic knowledge, unified embedding spaces combining acoustic and symbolic modalities, and deeper analysis of relationships between learned vector spaces and established linguistic taxonomies. The proposed methods and resulting models are intended to support further advances in multilingual speech technology and language documentation.

Acknowledgments. The authors thank Novosibirsk State University for support and provision of computational resources.

Conflict of Interest. The authors declare no competing interests.

References

1. eSpeak NG: Open source speech synthesizer (2025). <https://github.com/espeak-ng/espeak-ng>
2. Ardila, R., et al.: Common voice: a massively-multilingual speech corpus. arXiv preprint [arXiv:1912.06670](https://arxiv.org/abs/1912.06670) (2019)
3. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. In: Advances in Neural Information Processing Systems, vol. 33, pp. 12449–12460 (2020)
4. Benzeghiba, M., et al.: Automatic speech recognition and speech variability: a review. *Speech Commun.* **49**(10–11), 763–786 (2007)
5. Besacier, L., Barnard, E., Karpov, A., Schultz, T.: Automatic speech recognition for under-resourced languages: a survey. *Speech Commun.* **56**, 85–100 (2014)
6. Cheng, S., Liu, Z., Li, L., Tang, Z., Wang, D., Zheng, T.F.: ASR-free pronunciation assessment. arXiv preprint [arXiv:2005.11902](https://arxiv.org/abs/2005.11902) (2020)
7. Collobert, R., Puhersch, C., Synnaeve, G.: Wav2letter: an end-to-end convnet-based speech recognition system. arXiv preprint [arXiv:1609.03193](https://arxiv.org/abs/1609.03193) (2016)
8. Dong, L., Xu, S., Xu, B.: Speech-transformer: a no-recurrence sequence-to-sequence model for speech recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5884–5888. IEEE (2018)
9. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning, pp. 369–376 (2006)
10. Graves, A., Mohamed, A., Hinton, G.: Speech recognition with deep recurrent neural networks. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6645–6649. IEEE (2013)

11. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) SIMBAD 2015. LNCS, vol. 9370, pp. 84–92. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24261-3_7
12. Kingma, D.P.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
13. Li, B., Li, J., Zhao, R., Wang, Y., Gong, Y., Acero, A.: Universal phone recognition with a multilingual allophone system. In: ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8249–8253. IEEE (2020). <https://doi.org/10.1109/ICASSP40776.2020.9053564>
14. Li, Y.A., Han, C., Jiang, X., Mesgarani, N.: Phoneme-level BERT for enhanced prosody of text-to-speech with grapheme predictions. In: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5. IEEE (2023)
15. Lux, F., Vu, N.T.: Language-agnostic meta-learning for low-resource text-to-speech with articulatory features. arXiv preprint [arXiv:2203.03191](https://arxiv.org/abs/2203.03191) (2022)
16. McInnes, L., Healy, J., Melville, J.: Umap: uniform manifold approximation and projection for dimension reduction. arXiv preprint [arXiv:1802.03426](https://arxiv.org/abs/1802.03426) (2018)
17. Miao, Y., Gowayyed, M., Metze, F.: Eesen: end-to-end speech recognition using deep RNN models and WFST-based decoding. In: 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 167–174. IEEE (2015)
18. Mortensen, D.R., Littell, P., Bharadwaj, A., Goyal, K., Dyer, C., Levin, L.: Panphon: a resource for mapping IPA segments to articulatory feature vectors. In: Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pp. 3475–3484 (2016)
19. Novak, J.R., Minematsu, N., Hirose, K.: WFST-based grapheme-to-phoneme conversion: open source tools for alignment, model-building and decoding. In: Proceedings of the 10th International Workshop on Finite State Methods and Natural Language Processing, pp. 45–49 (2012)
20. Nowakowski, K., Ptaszynski, M., Murasaki, K., Nieuważny, J.: Adapting multilingual speech representation model for a new, underresourced language through multilingual fine-tuning and continued pretraining. *Inf. Process. Manag.* **60**(2), 103148 (2023)
21. Rabiner, L.R.: A tutorial on hidden Markov models and selected applications in speech recognition. In: Readings in Speech Recognition, pp. 267–296 (1990)
22. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLevey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: International Conference on Machine Learning, pp. 28492–28518. PMLR (2023)
23. Sanabria, R., Metze, F.: Hierarchical multitask learning with CTC. In: 2018 IEEE Spoken Language Technology Workshop (SLT), pp. 485–490 (2018). <https://api.semanticscholar.org/CorpusID:61807503>
24. Sanh, V., Wolf, T., Ruder, S.: A hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 6949–6956 (2019)
25. Sigtia, S., Marchi, E., Kajarekar, S., Naik, D., Bridle, J.: Multi-task learning for speaker verification and voice trigger detection. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6844–6848. IEEE (2020)
26. Smith, L.N.: Cyclical learning rates for training neural networks. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 464–472. IEEE (2017)

27. Snyder, D., Garcia-Romero, D., Sell, G., McCree, A., Povey, D., Khudanpur, S.: Speaker recognition for multi-speaker conversations using x-vectors. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5796–5800. IEEE (2019)
28. Toshniwal, S., et al.: Multilingual speech recognition with a single end-to-end model. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4904–4908. IEEE (2018)
29. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30, no. 1, pp. 261–272 (2017)
30. Xu, Q., Baevski, A., Auli, M.: Simple and effective zero-shot cross-lingual phoneme recognition. arXiv preprint [arXiv:2109.11680](https://arxiv.org/abs/2109.11680) (2021)
31. Yen, H.T., Wu, W.C., Chang, S.H., Tsao, Y., Wang, H.M., Hsieh, T.H.: Boosting end-to-end multilingual phoneme recognition through exploiting universal speech attributes constraints. arXiv preprint [arXiv:2309.08828](https://arxiv.org/abs/2309.08828) (2023)
32. Zhang, Y., Yang, Q.: A survey on multi-task learning. IEEE Trans. Knowl. Data Eng. **34**(12), 5586–5609 (2021)



Whistler Identification in Whistled Spanish (Silbo): A Case Study

Alejandro López-García¹ , María Alfaro-Contreras¹ , Julien Meyer² ,
and Jose J. Valero-Mas¹

¹ Pattern Recognition and Artificial Intelligence Group, University of Alicante,
Alicante, Spain

`alg166@gcloud.ua.es`, `{malfaro,jjvalero}@dlsi.ua.es`

² Université Grenoble Alpes, CNRS, GIPSA-Lab, Grenoble, France
`julien.meyer@cnrs.fr`

Abstract. Deemed one of the world’s most representative whistled languages, the Canary Islands’ whistled Spanish, locally known as Silbo, has long attracted linguistic research. However, most studies have adopted linguistic, ethnological, or bioacoustic perspectives, overlooking the potential of computational methods within the digital humanities. This work advances the computational study of Silbo by presenting the first automated approach to Speaker Identification (SI)—*i.e.*, the process of determining the speaker of a given utterance by computational means—in a closed-set configuration for this language. The proposal leverages standard feature extraction methods as well as pre-trained Speech Recognition models to extract representative embeddings and incorporates class-balancing mechanisms to mitigate biases arising from uneven representation of whistlers in the data—*i.e.*, label imbalance. The results obtained on the only existing dataset specifically designed for computational analysis of Silbo, comparing three representative feature extraction methods, three oversampling policies, and five classification strategies, validate the proposal, achieving F₁ scores close to 90% in the best-case scenarios. While laying a solid foundation for SI in Silbo, this study also highlights the scarcity of computational research on whistled languages, and particularly Silbo, emphasizing the need for further work to bridge traditional linguistic research and modern digital humanities.

Keywords: Whistled languages · Speaker identification · Silbo · Speech processing

1 Introduction

Whistled languages serve as a unique means of communication, in which regular spoken speech is replaced by modulated whistles that retain the same linguistic content while still enabling high intelligibility levels [20]. These languages are particularly useful in environments where spoken speech is ineffective due to challenging orographic conditions, such as long distances or dense jungles [3].

Nowadays, approximately 80 languages are known to have developed this speech type, but far fewer are regularly used worldwide [19].

The whistled Spanish of the Canary Islands, locally known as Silbo, stands as the most widely used whistled language in the world [32]. This prominence owes much to sustained promotional efforts by the regional government and cultural organizations, such as the “Asociación Cultural y de Investigación de lenguajes silbados Yo Silbo”¹, and to the inclusion of one of its variants (Silbo of La Gomera) on UNESCO’s Representative List of the Intangible Cultural Heritage of Humanity in 2009.²

Due to its cultural significance, Silbo has been a subject of research, including geolinguistic analyses of whistled speech [21], works on bioacoustics comparing whistles with communication among animals [22], and a large panel of psycholinguistic investigations, including studies exploring the relationship between musical knowledge and language proficiency [27] or comprehension analyses among expert practitioners of the tradition [23]. Nevertheless, these works have thereby neglected other aspects of the speech field, such as the computational one. It must be remarked that, while the related literature comprises seminal works related to low-level procedures for extracting signal-based descriptors from whistled speech [14], the study by Jakubiak [13] on the automated transcription of Silbo constitutes the first proposal focusing on the high-level analysis of this language by computational means. More recently, O’Brien and Marczyk [28] considered the data assortment presented in this latter work to differentiate modal and whistled speech with computational approaches.

In this work, we further contribute to the study of whistled Spanish from a computational perspective. More precisely, we present the first approach to Speaker Identification (SI) applied to Silbo, *i.e.*, the process of identifying the speaker—in our case, the whistler—of a given utterance through computational means [12]. Notably, this task proves to be remarkably relevant for the computational analysis of speech signals since, beyond biometric purposes [25], SI may enhance the accuracy of multi-speaker transcription systems by adapting the recognition framework to the characteristics of each individual speaker [2].

Our SI proposal for whistled Spanish leverages standard feature extraction methods as well as pre-trained Speech Recognition models to extract representative embeddings from the utterances, which are then post-processed and adapted for their eventual identification using classification systems. The results obtained using different state-of-the-art transcription models, data-balancing methods, and classification strategies on the Jakubiak dataset [13] prove the validity of our approach, achieving remarkably competitive classification performance in particular configurations. These findings support the effectiveness of the proposed method and lay the groundwork for future research in this field, contributing to the broader effort of preserving and raising awareness of these unique forms of communication.

¹ <http://www.yosilbo.com>.

² <https://ich.unesco.org/en/RL/whistled-language-of-the-island-of-la-gomera-canary-islands-the-silbo-gomero-00172>.

The remainder of this manuscript is structured as follows: Sect. 2 introduces the recognition framework developed for this task; Sect. 3 describes the experimental setup; Sect. 4 presents and discusses the obtained results; and finally, Sect. 5 concludes the work and outlines potential future research directions.

2 Methodology

This section formalizes the Speaker Identification (SI) problem and presents the methodology proposed to address this task. Note that, in this work, the SI problem is modeled as a multiclass classification task in which a query utterance must be identified as produced by one of the whistlers from a fixed set of candidates, *i.e.*, a closed-set identification framework.

Let \mathcal{X} and \mathcal{C} respectively denote the spaces of Silbo recordings and their associated labels (*i.e.*, the actual whistlers of the utterances), related by the function $\Omega : \mathcal{X} \rightarrow \mathcal{C}$. The goal of the SI task is to approximate this function as accurately as possible by learning an estimate $\hat{\Omega}$ using a set of labeled data, $\mathcal{T} \subset \mathcal{X} \times \mathcal{C}$. To achieve this, we propose the scheme illustrated in Fig. 1, which is described below.

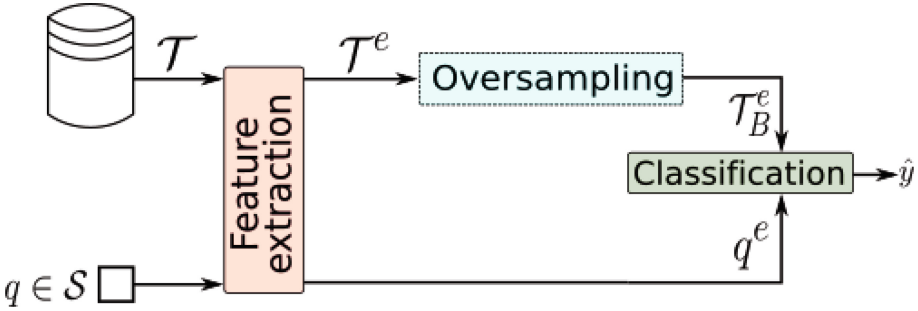


Fig. 1. Graphical description of the scheme proposed for the Speaker Identification task for Silbo speech.

The labeled Silbo recordings, $\mathcal{T} = \{(x_i, y_i) : x_i \in \mathcal{X}, y_i \in \mathcal{C}\}_{i=1}^{|\mathcal{T}|}$, initially undergo a feature extraction or embedding stage—denoted as the *Feature extraction* phase in the figure—resulting in the set $\mathcal{T}^e = \{(x_i^e, y_i) : x_i^e \in \mathbb{R}^f\}_{i=1}^{|\mathcal{T}^e|}$ of embedded data, where f denotes the size of the feature representation. To mitigate potential biases, the embedded representations \mathcal{T}^e are then artificially balanced in the *Oversampling* process, producing the adjusted set \mathcal{T}_B^e .³ The final

³ The limited representation of some whistlers in the considered Silbo assortment, which constitutes the only data collection of its type and that will be described in Sect. 3.1, prevents the use of balancing strategies based on undersampling procedures.

Classification stage then utilizes this balanced set to estimate the aforementioned function $\hat{\Omega}$.

During the inference phase, a query utterance q is drawn from a test set \mathcal{S} , which is disjoint from the training set \mathcal{T} —*i.e.*, $\mathcal{T} \cap \mathcal{S} = \emptyset$. The query q is processed by the *Feature extraction* phase, yielding the embedded representation $q^e \in \mathbb{R}^f$. Finally, the *Classification* method predicts the label of this query as $\hat{y} = \hat{\Omega}(q^e)$.

3 Experimental Set-Up

This section describes the data collection and evaluation protocol used to assess the proposed approach, as well as the embedding procedures, oversampling techniques, and classification strategies considered.

3.1 Data Assortment and Evaluation Protocol

We utilize the Silbo dataset compiled by Jakubiak [13], which represents the only existing assortment for the computational analysis of this language. This data collection comprises 529 whistled phrases recorded by 10 different practitioners, annotated at both word and sentence levels for transcription tasks. Table 1 provides further details on this collection in terms of the number of samples, total duration and average duration per sample.

Table 1. Details of the Silbo dataset compiled by Jakubiak [13] in terms of the number of samples, total duration and average duration per sample.

Number of Samples	Total Duration	Average Duration
529	1h 2 m 3.3 s	7.0 ± 2.9 s

For the SI task, we exclusively consider the practitioner labels provided in the dataset as the target elements, disregarding all transcription-related annotations. Figure 2 illustrates the distribution of speaker identifiers in this dataset.

Although Jakubiak [13] proposed a partitioning scheme, it was specifically designed for transcription purposes and does not account for the distribution of practitioners across different partitions. To address this, our experiments adopt a 5-fold cross-validation scheme, stratified at the practitioner level. Within each fold, 10% of the training samples are set aside for validation. Note that this partitioning scheme results in a closed-set identification configuration in which the whistler to be identified is among the set of reference practitioners [33]. Open-set scenarios are posed as future work to be explored.

Regarding the evaluation protocol, we use the macro-average F_1 score to measure the goodness of the proposal as it represents a standard figure of merit for

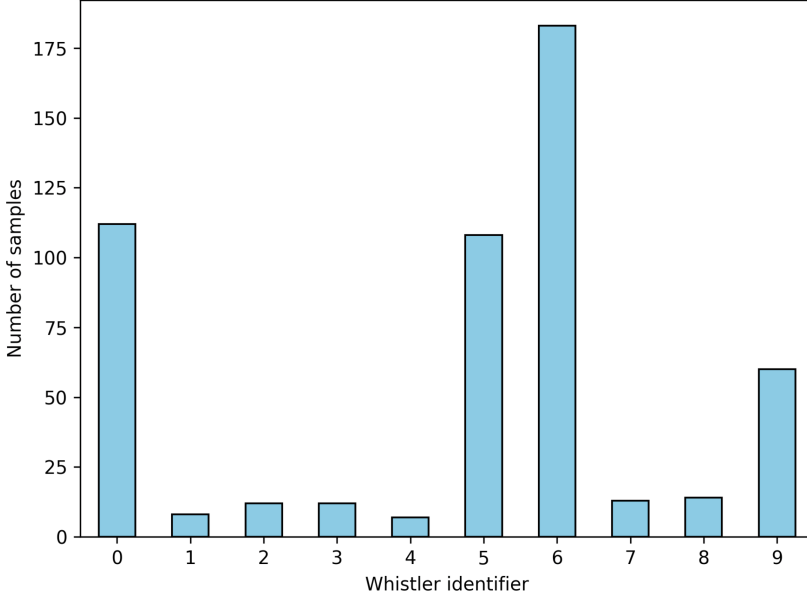


Fig. 2. Speaker identifier distribution of the Jakubiak dataset [13].

identification tasks [12] in contrast to other metrics more suitable for verification schemes such as Equal Error Rate [24]. The F_1 score is defined as:

$$F_1 = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} \frac{2 \cdot TP_c}{2 \cdot TP_c + FP_c + FN_c} \quad (1)$$

where TP_c , FP_c , and FN_c denote the True Positives, False Positives, and False Negatives for whistler identifier $c \in \mathcal{C}$, respectively.

3.2 Feature Extraction

Given their competitive performance in the speech processing field, we adopt and compare three different feature extraction strategies for the identification task: the spectral-based mel-frequency cepstral coefficients (MFCC) commonly considered for speech analysis [8] together with two neural embedding schemes based on Speech Recognition models, *wav2vec 2.0* by Meta [1] and *Whisper* [31] by OpenAI. The remainder of this section provides a brief overview of these methods.

The *MFCC* descriptors constitute a set of low-level features which directly represent the spectral content of the utterance at hand. More precisely, this representation results from processing the Fourier Transform of the signal with a filter bank based on the perceptual Mel scale. We initially consider 20 filters (namely *Base*) and examine the influence of including both their first ($Base + \Delta$) and second derivatives ($Base + \Delta + \Delta^2$), as commonly done in SI with regular

speech [24, 33]. These cases result in embedding sizes of $f = \{20, 40, 60\}$, respectively.

The *wav2vec 2.0* scheme consists of a multi-layer convolutional neural network, whose output embeddings are processed by a Transformer network for transcription. This model is pre-trained and fine-tuned on 960 h of the LibriSpeech dataset [29]. Note that we consider only the encoder component of the model as our objective is to obtain meaningful embedded representations of Silbo utterances. To evaluate its impact on the overall performance of our SI task, we experiment with different embedding sizes for the encoder, specifically $f = \{64, 256, 1024, 4096\}$.

The *Whisper* model is an end-to-end speech transcription system based on an encoder-decoder Transformer architecture. It is trained on 680,000 h of multilingual and multitask supervised data collected from the Internet. As in the previous scheme, we exclusively consider the encoder stage of the scheme as it acts as a feature extractor for our task. In our experiments, we consider four versions of the model, which differ in the number of trainable parameters: *Tiny* (39M), *Base* (64M), *Small* (244M), and *Medium* (769M). The corresponding embedding sizes for these encoders are $f = \{384, 512, 768, 1024\}$, respectively.

3.3 Oversampling Techniques

To mitigate the effects of the label imbalance present in the dataset, we employ three well-known oversampling strategies from the literature [26]: (i) Synthetic Minority Over-Sampling Technique (SMOTE) [4], (ii) Borderline SMOTE (B-SMOTE) [10], and (iii) Adaptive Synthetic Sampling (ADASYN) [11]. Since these methods require a feature-based representation of the data, the oversampling process is applied after the *Feature extraction* stage.

The SMOTE technique addresses class imbalance by generating synthetic samples in the regions of the \mathbb{R}^f feature space occupied by the minority classes. More precisely, the algorithm first selects a random sample from a minority class as well as one of its nearest neighbors in a random manner. A new synthetic sample is then generated by interpolating between the reference sample and the selected elements, with the new instance inheriting the reference sample's label. This process is repeated for each class until a predefined balancing criterion is met (*e.g.*, ensuring all classes contain the same number of instances).

The B-SMOTE method extends the SMOTE algorithm by focusing on decision boundaries between classes. The oversampling process follows the same steps as SMOTE, with the key distinction that reference samples are specifically chosen from those lying on the decision frontiers—*i.e.*, instances predominantly surrounded by samples from the majority class.

The ADASYN algorithm differs from SMOTE-based strategies by employing an adaptive generation policy that prioritizes minority instances that are more difficult to classify, rather than uniformly sampling the minority classes. To achieve this, ADASYN uses a set of indicators to assess classification difficulty in terms of label imbalance and generates synthetic samples accordingly to reduce these disparities.

3.4 Classification Strategies

Regarding the *Classification* stage of the proposal, we examine five representative methods from the literature [7] with large application in the field of computational speech analysis, which are listed and described in the remainder of the section. Note that, for each classifier we assess different configuration parameters to optimize their recognition performance for the proposed SI task.

Based on its success in speech processing tasks, we examine the Gaussian Mixture Model (GMM) as a representative case of parametric learning [6]. For its use as a classifier, we fix the amount of gaussian functions in the mixture to the number of whistlers in the dataset. We initialize the centers of the distributions to those of the respective whistlers and optimize the rest of the parameters via Expectation-Maximization. We comparatively study the influence of the covariance function by comparing the case in which all components share the same general covariance matrix against that in which each component has its own single variance, respectively denoted as *tied* and *spherical* (SPH) in the rest of the work.

As a representative of the lazy learning paradigm, we consider the k -Nearest Neighbor (k NN) method, which classifies a given query based on the labels of the k elements that surround it in the feature space [35]. To assess the impact of the k hyperparameter on SI classification performance, we evaluate the method using $k \in \{1, 3, 5\}$.

In terms of neural networks, we explore the Multilayer Perceptron (MLP) [9] scheme as an example of this learning family. In this case, we analyze the effect of the optimization strategy by comparing classification performance when using the Limited-memory Broyden–Fletcher–Goldfarb–Shannon (LBFGS) algorithm versus the Stochastic Gradient Descent (SGD) method.

Regarding tree-based strategies, we evaluate the Random Forest (RaF) [15] method, which typically outperforms individual decision tree classifiers by leveraging an ensemble of such base classifiers to enhance robustness and reduce overfitting. To examine the influence of the number of trees in the ensemble, we test configurations with 100 and 500 trees.

Finally, given its competitive performance in related literature, we also include the Support Vector Machine (SVM) classifier in our study [17]. On this note, we compare two commonly used kernel functions: the polynomial one (Poly) and the Radial Basis Function (RBF).

4 Results

This section presents and discusses the results obtained for the SI task, following the experimental procedure outlined in Sect. 3. For clarity, the analysis is divided into two parts: (i) a comparative evaluation of feature extraction and classification methods, focusing on base SI performance without applying balancing strategies, and (ii) an assessment of the impact of oversampling techniques on mitigating class imbalance to enhance the overall SI performance.

To ensure reproducibility and transparency, all developed code is publicly available at <https://github.com/jose-jvm/WhistlerIdentificationSilbo>. All experiments were conducted using Python (v. 3.11) with the *Hugging Face Transformers* (v. 4.46.3) [34], *librosa* [18], *scikit-learn* (v. 1.6.1) [30], and *imbalanced-learn* (v. 0.13.0) [16] libraries for the feature extraction, classification, and evaluation tasks. Finally, Table 2 summarizes the different experimental parameters assessed in the work.

Table 2. Summary of the experimental parameters considered in the experimentation categorized by those related to the embedding strategies, the classification methods, and the oversampling approaches.

Parameter	Value
<i>Feature extraction</i>	
MFCC	Base, Base + Δ , Base + Δ + Δ^2
Wav2vec	64, 256, 1024, 4096
Whisper	Tiny, Base, Small, Medium
<i>Classification methods</i>	
Gaussian Mixture Models (GMM)	Tied, Spherical
k -Nearest Neighbor (k NN)	1, 3, 5
Multi-Layer Perceptron (MLP)	Stochastic Gradient Descent Lim. Broyden–Fletcher–Goldfarb–Shannon
Random Forest (RaF)	100, 500
Support Vector Machine (SVM)	Polynomial, Radial Basis Function
<i>Oversampling approaches</i>	
Methods	Synthetic Minority Over-Sampling Technique Borderline-SMOTE Adaptive Synthetic Sampling

4.1 Base Identification Performance

Table 3 presents the F_1 score performance of the proposed SI scheme across different feature extraction and classification strategies. Note that this initial analysis does not incorporate any balancing methods.

Overall, MFCC-based representations consistently achieve the highest identification rates across all classifiers. The best result, an F_1 score of 87.9%, is obtained using an SVM classifier with a polynomial kernel and MFCC features augmented with first-order derivatives (Base + Δ). This finding aligns with the speaker recognition literature, confirming the effectiveness of MFCCs for distinguishing individual whistlers in Silbo, likely due to the prominent spectral patterns in whistled speech.

Table 3. Average test results for the 5-fold cross-validation scheme in terms of F_1 (%) for the classifiers evaluated with respect to the embedding strategy without oversampling. Bold values highlight the best result for each classification scheme and embedding method, while underlined values indicate the best overall result per classifier. Feature sizes (f) are provided for comparison.

	GMM		k NN			MLP		RaF		SVM	
	SPH	Tied	1	3	5	LBFGS	SGD	100	500	Poly	RBF
<i>MFCC</i>											
Base (20)	24.5	72.7	75.2	70.5	68.8	80.5	23.4	68.8	66.7	85.8	22.2
Base + Δ (40)	22.3	79.3	76.2	70.9	70.3	82.5	7.7	64.9	66.1	87.9	22.1
Base + Δ + Δ^2 (60)	19.1	78.0	75.6	67.7	67.4	78.8	16.9	66.2	68.3	86.3	22.2
<i>Wav2vec</i>											
64	8.0	9.6	12.2	12.0	14.0	10.7	12.6	10.2	10.3	12.5	10.7
256	7.5	11.3	12.3	12.2	13.9	11.7	12.0	11.9	11.8	13.8	13.5
1024	6.5	3.6	11.3	10.4	10.1	12.3	11.9	13.2	13.7	12.7	13.8
4096	6.8	1.9	12.4	10.1	8.7	16.0	14.7	12.0	11.6	13.0	11.0
<i>Whisper</i>											
Tiny (384)	7.1	49.5	39.6	37.9	29.2	57.1	7.8	26.3	23.4	83.7	11.7
Base (512)	10.6	65.4	47.4	38.7	36.9	45.2	5.1	34.7	36.2	83.8	11.0
Small (768)	7.7	23.0	26.3	20.7	19.3	31.2	5.1	21.7	21.4	71.3	9.1
Medium (1024)	12.8	59.0	34.3	35.4	27.5	34.4	4.5	32.2	30.2	74.5	7.4

Focusing on the particular MFCC configurations, it is observed that incorporating first-order derivatives (Base + Δ) generally improves performance compared to using only the base coefficients, suggesting that the inclusion of dynamic spectral information benefits the discrimination between whistlers. Interestingly, the best identification rates are achieved with the first-order configuration (Base + Δ) across all classifiers except for RaF, which performs best with the base coefficients alone.

In contrast, Wav2vec embeddings yield the lowest performance across classifiers, with F_1 scores typically below 15%. This suggests that features learned from spoken speech do not generalize well to whistled signals within the SI context. Conversely, Whisper-based embeddings show improved performance, achieving F_1 scores up to 83.8% with the SVM classifier. This indicates that large-scale multilingual pre-training confers better generalization to non-verbal speech modalities such as whistling. Among the Whisper configurations, model complexity plays an important role, with the best results obtained using the less complex Tiny and Base models.

Finally, the choice of classification scheme plays a critical role in the SI task, with different methods showing varying sensitivities to parameter tuning. The SVM classifier with a polynomial kernel consistently delivers the highest performance across feature types, highlighting its capacity to generalize effectively in

this context. Tree-based methods (RaF) and k NN classifiers also provide stable yet slightly lower performance with minimal tuning. In contrast, methods such as MLP and GMM can achieve competitive results but require careful configuration to reach optimal performance.

In summary, the results obtained establish a strong baseline for speaker identification in Silbo, demonstrating that classical spectral features based on cepstral principles combined with well-tuned classifiers can achieve high recognition rates even in label-imbalance cases.

4.2 Oversampling Strategies for Data Balancing

Following the initial analysis, this section examines the impact of oversampling techniques on addressing label imbalance within the Silbo dataset. For conciseness, we focus on the best-performing classifier configurations identified in Sect. 4.1: GMM with Tied covariance, k NN with $k = 1$, MLP with the LBFGS optimizer, RaF with 500 trees, and SVM with the Poly kernel. Additionally, we restrict the analysis to the MFCC and Whisper feature extraction methods under the configurations yielding the highest performance for each classifier.

Table 4 presents the F_1 scores obtained for the proposed SI scheme when applying different oversampling strategies. The None case—*i.e.*, no balancing method is applied—serves as the baseline for comparison.

Table 4. Average test results for the 5-fold cross-validation scheme in terms of F_1 (%) for the best classification configurations from Table 3, evaluated across different oversampling strategies. Bold values indicate the best result per embedding method, classification scheme, and oversampling strategy, while underlined values highlight the best overall score per embedding method.

	Oversampling method			
	None	SMOTE	B-SMOTE	ADASYN
<i>MFCC</i>				
GMM	79.3	78.5	79.3	81.7
k NN	76.2	79.6	78.2	79.0
MLP	82.5	72.1	81.3	69.0
RaF	68.8	73.2	73.3	73.5
SVM	<u>87.9</u>	86.2	85.8	86.2
<i>Whisper</i>				
GMM	65.4	72.2	56.2	62.7
k NN	47.4	53.9	51.4	54.0
MLP	57.1	68.7	70.2	66.3
RaF	36.2	58.5	53.3	59.2
SVM	83.8	83.7	83.5	<u>85.2</u>

The results indicate that applying oversampling generally improves classification performance, although the magnitude of enhancement depends on both the feature representation and the baseline performance without balancing (*i.e.*, the None case). For MFCC features, increases in the figure of merit are observed primarily for the GMM, k NN, and RaF classifiers, while MLP and SVM do not benefit to the point of achieving their best performance without oversampling. For instance, RaF improves from 68.8% to 73.5% F_1 with ADASYN, and k NN increases from 76.2% to 79.6% with SMOTE, while SVM maintains its highest performance (87.9% F_1) without balancing, with only marginal changes across oversampling methods.

In the case of Whisper-based features, oversampling consistently yields improvements across k NN, MLP, and RaF, indicating that balancing is particularly beneficial when the feature representation alone does not ensure sufficient class discrimination. Notably, RaF improves from 36.2% to 59.2% F_1 with ADASYN, and MLP increases from 57.1% to 70.2% with B-SMOTE. SVM, while already performing well with Whisper features, shows a slight improvement from 83.8% to 85.2% F_1 with ADASYN. GMM, in contrast, only benefits with SMOTE, improving from 65.4% to 72.2%, but exhibits lower results with other oversampling strategies.

Among the oversampling methods evaluated, ADASYN consistently provides the highest or near-highest performance across multiple configurations. This aligns with its adaptive sampling policy, which prioritizes generating synthetic examples in regions where minority classes are harder to classify, unlike SMOTE-based methods that distribute synthetic samples more uniformly.

Within the SMOTE family, no clear overall advantage emerges between SMOTE and B-SMOTE, with their relative performance highly depending on the classifier-feature configuration. For example, B-SMOTE slightly outperforms SMOTE with MLP on Whisper features, while the inverse is true for k NN.

Overall, the best results across all configurations are achieved using the SVM classifier with MFCC features, maintaining an F_1 score of 87.9% without requiring oversampling. For Whisper features, the highest F_1 score (85.2%) is achieved with SVM using ADASYN, illustrating the potential of oversampling to close the performance gap between Whisper and MFCC representations under optimal configurations.

To conclude the analysis, we conduct a Nemenyi post-hoc test [5] to assess the statistical significance of differences among classifiers. Figure 3 shows the results for both MFCC and Whisper features, considering a significance threshold of $p < 0.05$.

As it can be observed, the analysis confirms the similar performance of the k NN, RaF, and MLP classifiers under many configurations, while the SVM consistently ranks highest. This fact reinforces its superior generalization capacity for SI in Silbo, as previously observed.

Finally, we conclude that while oversampling can remarkably enhance performance in label-imbalanced and low-separability settings, high-performing configurations such as MFCC combined with SVM do achieve strong results without

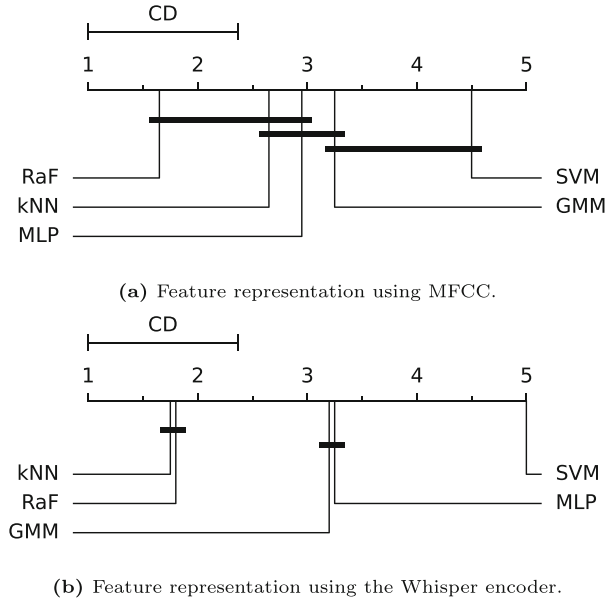


Fig. 3. Results of the Nemenyi post-hoc test assessing the relative improvement across the different classification strategies for the MFCC and Whisper representations depicting the most competitive results. The Critical Distance (CD) represents the minimum difference in the significance score to consider a pair of classifiers as statistically different.

requiring data balancing. This establishes a solid benchmark for future computational studies on whistled language SI under data-scarce scenarios.

5 Conclusions and Future Work

The whistled Spanish of the Canary Islands, or Silbo, is one of the world’s most representative whistled languages and has long been the focus of a number of linguistic studies. As the most intensively studied form of whistled speech worldwide, Silbo has yielded insights into ethnological, linguistic, and cognitive phenomena, but computational methodologies for advanced, automated analysis remain largely unexplored.

This work presents the first approach to Speaker Identification (SI)—*i.e.*, the process of determining the speaker of a given utterance via computational means—tailored to Silbo in a closed-set disposition. Leveraging a combination of standard spectral-based feature extraction methods, embeddings from pre-trained Speech Recognition models, and class-balancing techniques, we prove that automatic SI for Silbo can successfully be carried out with competitive performance rates under data-scarce conditions.

Extensive experiments conducted on the Jakubiak dataset [13]—the only resource specifically compiled for computational analysis of Silbo—show that

MFCC-based features, particularly when combined with an SVM classifier using a polynomial kernel, deliver the best performance, achieving an F_1 score of 87.9%. Whisper-based embeddings also demonstrate competitive results, reaching up to 85.2% F_1 with oversampling, although their effectiveness is highly dependent on the classification strategy considered. In contrast, embeddings derived from Wav2vec exhibit limited applicability in this context, highlighting the challenges of directly transferring features learned from monolingual spoken speech to whistled modalities.

Our findings highlight that, while oversampling techniques can remarkably enhance performance in scenarios with lower class separability (e.g., Whisper-based features), high-performing configurations such as MFCC with SVM can achieve strong results without requiring data balancing. This establishes a robust baseline for computational SI in Silbo, demonstrating the potential of standard feature representations and classifiers in addressing this underexplored task. However, despite these promising results, the scarcity of whistled data severely hinders progress in this field, which highlights the need for larger datasets.

Based on the above, future work will focus on expanding existing data collection to include a larger and more diverse collection of practitioners, whistling styles, and recording conditions. Other aspects to be explored comprise the adaptation or fine-tuning of neural-based embedding models for extracting adequate representations for whistled speech, or the use of data-efficient approaches such as Siamese networks for low-resource SI scenarios. Additionally, the integration of multimodal representations, combining spectral, temporal, and possibly articulatory features, may further enhance system robustness and accuracy. Extending this work to open-set SI, speaker verification, and speaker diarization tasks in real-world whistled speech recordings is also a promising direction for future exploration.

All in all, our work lays a solid foundation for the computational analysis of Silbo, showing the feasibility and effectiveness of SI for whistled speech and bridging the gap between traditional linguistic research and modern machine learning techniques within the digital humanities.

Acknowledgments. This work was partially funded by the Generalitat Valenciana through project CIGE/2023/216.

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. In: *Advances in Neural Information Processing Systems*, vol. 33, pp. 12449–12460 (2020)
2. Bai, Z., Zhang, X.L.: Speaker recognition based on deep learning: an overview. *Neural Netw.* **140**, 65–99 (2021)
3. Busnel, R.G., Classe, A.: *Whistled Languages*, vol. 13. Springer, Cham (2013)
4. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: Smote: synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357 (2002)

5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **7**, 1–30 (2006)
6. Dhanjal, A.S., Singh, W.: A comprehensive survey on automatic speech recognition using neural networks. *Multimed. Tools Appl.* **83**(8), 23367–23412 (2024)
7. Duda, R.O., Hart, P.E., et al.: *Pattern Classification*. Wiley, Hoboken (2006)
8. Ganchev, T., Fakotakis, N., Kokkinakis, G.: Comparative evaluation of various MFCC implementations on the speaker verification task. In: *Proceedings of the International Conference of Speech and Computer (SPECOM)*, vol. 1, pp. 191–194 (2005)
9. Han, B., Chen, Z., Liu, B., Qian, Y.: MLP-svnet: a multi-layer perceptrons based network for speaker verification. In: *International Conference on Acoustics, Speech and Signal Processing*, pp. 7522–7526 (2022)
10. Han, H., Wang, W.Y., Mao, B.H.: Borderline-smote: a new over-sampling method in imbalanced data sets learning. In: *International Conference on Intelligent Computing*, pp. 878–887 (2005)
11. He, H., Bai, Y., Garcia, E.A., Li, S.: Adasyn: adaptive synthetic sampling approach for imbalanced learning. In: *International Joint Conference on Neural Networks*, pp. 1322–1328 (2008)
12. Jahangir, R., Teh, Y.W., Nweke, H.F., Mujtaba, G., Al-Garadi, M.A., Ali, I.: Speaker identification through artificial intelligence techniques: a comprehensive review and research challenges. *Expert Syst. Appl.* **171**, 114591 (2021)
13. Jakubiak, A.: Whistle-to-text: automatic recognition of the silbo gomero whistled language. In: *Proceedings of the 24th INTERSPEECH Conference*, pp. 3402–3406 (2023)
14. Johansson, A.T., White, P.R.: An adaptive filter-based method for robust, automatic detection and frequency estimation of whistles. *J. Acoust. Soc. Am.* **130**(2), 893–903 (2011)
15. Karthikeyan, V., Suja Priyadharsini, S.: Adaptive boosted random forest-support vector machine based classification scheme for speaker identification. *Appl. Soft Comput.* **131**, 109826 (2022)
16. Lemaître, G., Nogueira, F., Aridas, C.K.: Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **18**(17), 1–5 (2017)
17. Malik, M., Malik, M.K., Mehmood, K., Makhdoom, I.: Automatic speech recognition: a survey. *Multimed. Tools Appl.* **80**, 9411–9457 (2021)
18. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: audio and music signal analysis in python. *SciPy* **2015**, 18–24 (2015)
19. Meyer, J.: Whistled languages. *A Worldwide Inquiry on Human Whistled Speech* (2015)
20. Meyer, J.: Environmental and linguistic typology of whistled languages. *Ann. Rev. Linguist.* **7**(1), 493–510 (2021)
21. Meyer, J., Díaz Reyes, D.: Geolingüística de los lenguajes silbados del mundo, con un enfoque en el español silbado. *Géolinguistique* **17**, 99–124 (2017)
22. Meyer, J., Magnasco, M.O., Reiss, D.: The relevance of human whistled languages for the analysis and decoding of dolphin communication. *Front. Psychol.* **12**, 689501 (2021)
23. Meyer, J., Rolland, V., Socas, T., Díaz, D.: A sentence comprehension test with whistled Spanish experts. In: *ExLing Conferences*, pp. 65–68 (2024)
24. Mittal, A., Dua, M.: Automatic speaker verification systems and spoof detection techniques: review and analysis. *Int. J. Speech Technol.* **25**(1), 105–134 (2022)

25. Mohd Hanifa, R., Isa, K., Mohamad, S.: A review on speaker recognition: technology and challenges. *Comput. Electr. Eng.* **90**, 107005 (2021)
26. Mujahid, M., et al.: Data oversampling and imbalanced datasets: an investigation of performance for machine learning and feature engineering. *J. Big Data* **11**(1), 87 (2024)
27. Ngoc, A.T., Meyer, J., Meunier, F.: The effect of musical expertise on whistled vowel identification. *Speech Commun.* **159**, 103058 (2024)
28. O'brien, B., Marczyk, A.: A spectrotemporal modulation application for distinguishing modal and whistled speech. *Int. J. Speech Technol.* 1–8 (2025)
29. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an ASR corpus based on public domain audio books. In: *International Conference on Acoustics, Speech and Signal Processing*, pp. 5206–5210 (2015)
30. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
31. Radford, A., Kim, J.W., Xu, T., Brockman, G., McLeavey, C., Sutskever, I.: Robust speech recognition via large-scale weak supervision. In: *Proceedings of the 40th International Conference on Machine Learning*, pp. 28492–28518 (2023)
32. Tapiador, F.J.: Heritage: a treasure chest. In: *The Geography of Spain: A Complete Synthesis*, pp. 405–419 (2020)
33. Tirumala, S.S., Shahamiri, S.R., Garhwal, A.S., Wang, R.: Speaker identification features extraction methods: a systematic review. *Expert Syst. Appl.* **90**, 250–271 (2017)
34. Wolf, T., et al.: Transformers: state-of-the-art natural language processing. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45 (2020)
35. Yerramreddy, D.R., et al.: Speaker identification using MFCC feature extraction: a comparative study using GMM, CNN, RNN, KNN and random forest classifier. In: *International Conference on Trends in Electrical, Electronics, and Computer Engineering*, pp. 287–292 (2023)

Digital Speech Processing



PinkVocalTransformer: Neural Acoustic-to-Articulatory Inversion Based on the Pink Trombone

Zhiyuan Xu^(✉) and Joshua Reiss

Centre for Digital Music, Queen Mary University of London, London, UK
{zhiyuan.xu,joshua.reiss}@qmul.ac.uk

Abstract. Articulatory synthesis generates speech by modeling vocal tract configurations, but estimating articulatory parameters from audio—the acoustic-to-articulatory inversion (AAI) problem—remains challenging due to data scarcity, ambiguity, and the limitations of optimization-based methods. We propose PinkVocalTransformer, a Transformer framework that reformulates AAI as a sequence-to-sequence classification task over 44-dimensional vocal tract diameter sequences derived from the Pink Trombone physical synthesizer. By modeling complete tract shapes rather than higher-level articulatory trajectories, our approach yields a more interpretable and spatially consistent representation. To enable supervised learning, we generated over four million synthetic audio–parameter pairs under controlled static configurations. HuBERT embeddings improve feature extraction and robustness to real audio inputs. Reformulating regression as classification helps mitigate convergence issues arising from multimodal parameter distributions, leading to more stable predictions. Since ground-truth articulatory data are unavailable for real recordings, we regenerate audio from predicted parameters to indirectly evaluate reconstruction quality. Experiments show PinkVocalTransformer outperforms VAE-based and optimization baselines in vowel reconstruction. Objective ViSQOL metrics and ABX listening tests confirm higher perceptual similarity and listener preference for the regenerated audio compared to baselines. While the model performs strongly on static and simple dynamic segments, future work will focus on extending coverage to more diverse articulatory transitions and adapting the framework to more complex vocal tract models. Overall, this approach provides an efficient, data-driven framework for recovering interpretable articulatory parameters from audio, demonstrating both improved reconstruction quality and perceptual similarity compared to existing baselines.

Keywords: Acoustic-to-articulatory inversion · Transformer · Articulatory synthesis · Pink trombone

1 Introduction

Articulatory synthesis [1] generates speech by simulating vocal tract dynamics. Unlike statistical [2, 3] or concatenative methods [4, 5], it explicitly models artic-

ulatory motion, offering better interpretability and control. These advantages benefit linguistic research and have potential to help diagnose speech disorders and vocal tract conditions.

Several foundational models have supported articulatory synthesis, including the Liljencrants-Fant (LF) model [6] and the source-filter theory [7], which offer key insights into speech mechanics. Building on these, physical vocal tract simulators such as Pink Trombone (PT) [8] and VocalTractLab [9] have been developed to study articulatory coordination.

Despite these advances, realistic synthesis remains difficult. Black-box methods rely on optimization but face local minima and high cost. White-box approaches infer parameters analytically but are also costly and inefficient at scale.

To address these challenges, we developed a black-box deep learning method to inversely model the shape of the vocal tract. As shown in Fig. 1, our approach takes acoustic signals as input and outputs the articulatory parameters that best reconstruct the original sound. To establish this mapping, we use PT as the synthesizer and generate a dataset of more than four million static audio parameter pairs. Unlike previous studies [10, 11], which often rely on traditional articulatory features, we focus on the diameters of the vocal tract as articulatory parameters, representing them as a 44-dimensional sequence [12]. Consequently, the problem can be framed as mapping acoustic representations onto a spatial sequence of articulatory diameters.

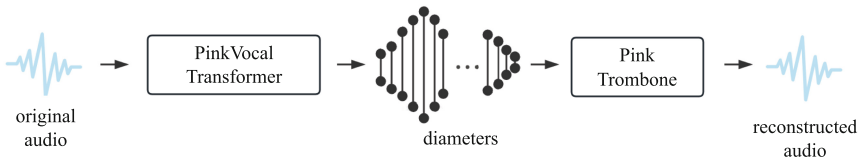


Fig. 1. Work process of the PinkVocalTransformer.

A major challenge in this approach is the instability caused by the multi-peak distribution of articulatory parameters, which complicates training. To address this, we reformulate the regression task as classification to stabilize learning. Since training only on PT data limits generalization to real-world audio, we integrate the pretrained HuBERT model [13] into the embedding layer to enhance feature extraction.

The objectives of this paper are the following.

- Propose a method that formulates acoustic-to-articulatory inversion (AAI) as a sequence-to-sequence problem and
- Evaluate the model’s accuracy and robustness through systematic testing on both synthetic and real audio.

The remainder of this paper is organized as follows. Section 2 reviews related work. Section 3 introduces the PT-based dataset and the proposed Transformer-based method. Section 4 presents experimental results, and Sect. 5 discusses and concludes the findings.

2 Related Work

Articulatory synthesis models the vocal tract and controls its motion to simulate articulatory behavior and generate audio. Early efforts include Kempelen’s 18th-century mechanical synthesizer [14] and the computational vocal tract model introduced by Kelly and Lochbaum in 1962 [15], which laid the foundation for digital articulatory simulation. With advances in medical imaging technologies, such as magnetic resonance imaging (MRI) and computed tomography (CT), and in modeling methods, researchers have integrated structures such as lips and tongue into simulations, greatly improving precision and expanding applications beyond audio generation.

As modeling techniques [6, 7, 16] evolved, early work focused on building paired datasets of audio and articulatory parameters, often using codebook-based inversion methods [17, 18]. Later, analysis-by-synthesis approaches [1–3] matched model-generated audio to targets using iterative optimization, but these methods were time-consuming and vulnerable to local minima [19], limiting accuracy and scalability.

Deep learning offers a more efficient alternative by directly learning the mapping between acoustic and articulatory features. Prior work typically used vocal tract parameters as targets [10, 19, 21], with acoustic features such as Mel spectrograms, MFCCs, and related variants. Models including convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and variational autoencoders (VAEs) have all been applied to this task.

Study [11], for example, introduced a two-head VAE architecture where one decoder reconstructs the mel-spectrogram while the other predicts six Pink Trombone control parameters. The authors also explored pretrained encoders such as EnCodec and wav2vec2.0 combined with lightweight projector networks for parameter estimation. While effective, these methods rely on low-dimensional control vectors and predict all target values in a single step, implicitly assuming conditional independence, which may limit the model’s ability to capture structured dependencies among articulatory positions.

In contrast, our method uses a full 44-dimensional sequence of vocal tract diameters as articulatory features, providing a more intuitive geometric representation. This formulation supports an autoregressive decoder that captures spatial dependencies across articulatory positions and produces more coherent reconstructions.

3 Datasets and Methods

This section describes our method for static acoustic-to-articulatory inversion (AAI), which maps short audio segments to vocal tract diameter sequences using

a Transformer-based architecture. We first introduce the Pink Trombone (PT) synthesizer and its articulatory parameterization, followed by the dataset construction process and modeling of glottal excitation. Finally, we explain the use of HuBERT-based feature extraction and the reformulation of regression into classification to improve training stability.

3.1 Pink Trombone

PT is a two-dimensional physical model of the human vocal tract that simulates audio production using a compact set of parameters, including constriction location, tongue location, and glottal excitation. While generating audio from these parameters is straightforward, the inverse problem remains computationally challenging. Prior PT-based studies [10] derived parameters from user interactions, with ranges listed in Table 1, but assumed tongue and constriction movements occur simultaneously, whereas they can occur independently. We address this by using 44-dimensional diameter sequences.

Table 1. Ranges of User Interaction Parameters

Parameters	Lower Bound	Upper Bound
pitch (Hz)	75	330
voiceness	0	1
tongue index	14	27
tongue diameter (cm)	1.55	3
lips diameter (cm)	0.6	1.2
constriction index	12	42
constriction diameter (cm)	0.6	1.2
throat diameter (cm)	0.5	1.0

3.2 Data

To construct the experimental dataset, we generated two types of audio based on the user interaction parameters in Table 1: one with constriction interactions and one without. Each sample was paired with its corresponding vocal tract diameter sequence, forming articulatory–acoustic mappings.

To ensure adequate parameter coverage, we used Latin Hypercube Sampling (LHS) [26], which divides each parameter’s range into intervals and randomly samples one value per interval. The resulting combinations were input into Pink Trombone to generate the corresponding audio signals.

The resulting dataset contains 4,374,000 pairs, including 4,252,500 with constrictions. To improve robustness under noise, we applied augmentation by adding white noise at signal-to-noise ratios (SNRs) of 10 and 5. This not only

enhances generalization in noisy conditions but also expands the dataset. The final version is denoted as *pt_data_exlarge*.

3.3 Glottal Flow Derivative

PT uses the LF model to generate glottal flow derivative (GFD) waveforms, represented by the parameter R_d , which correlates with perceived vocal effort [22] and can be estimated from the GFD spectrum [23]. PT does not directly use R_d as a control parameter, but instead adopts a related parameter, Tenseness (T), defined as $T = 1 - R_d/3$. The prediction of (T) follows the approach in [24], while the fundamental frequency is predicted using the CREPE model [25].

3.4 Pretrained Models

Without pretrained models, deep models trained solely on *pt_data_exlarge* perform well on PT reproduction but generalize poorly to real audio, consistent with prior findings [10, 20]. This limitation arises because the training data, which are entirely generated by PT, lack speaker variability. Consequently, the model cannot distinguish speaker-dependent characteristics from the underlying articulatory content when applied to real audio, regardless of whether Mel spectrograms or MFCCs are used.

To address this, we incorporate a pretrained model for feature extraction. Given the static nature and short duration (0.125s) of the audio, we opted against wav2vec2.0 [30], which relies on contrastive learning over long sequences and is better suited for dynamic audio tasks. In contrast, HuBERT uses unsupervised clustering-based self-supervised learning, making it more effective for capturing phonetic representations in short signals. Its ability to produce contextualized embeddings from limited temporal context enables better feature extraction for static AAI. While HuBERT embeddings improve robustness compared to conventional acoustic features, they still retain some speaker-dependent characteristics, which can introduce variability when applied to recordings with unseen speakers.

3.5 Regression to Classification

We initially implemented a Transformer-based regression model using conditional sequence modeling to predict 44 vocal tract diameters from acoustic features. Despite experimenting with both MSE and Huber loss, the model converged slowly and yielded suboptimal performance. Prior studies [27] have shown that Transformers often struggle with regression tasks due to error accumulation and high data demands. Additionally, standard loss functions that ignore dependencies among the 44 dimensions may further limit model effectiveness.

To address this, we reformulated the task as classification to better leverage Transformer architectures. We analyzed the parameter distributions with the Freedman–Diaconis rule [28], and used the resulting histogram bins as class intervals.

Although discretization introduces quantization error, it significantly improved performance. The multimodal nature of most diameter distributions further supports this approach, as classification can be viewed as a tokenization process that helps Transformers better model multimodal targets [29]. Ultimately, the 44 continuous diameter values were transformed into a classification task with 6,123 discrete categories.

3.6 PinkVocalTransformer

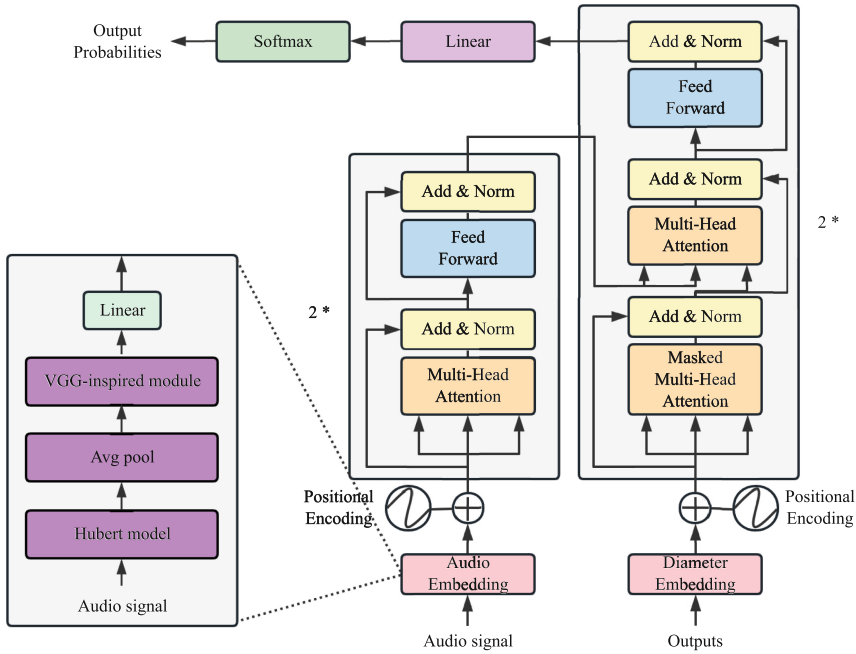


Fig. 2. Structure of PinkVocalTransformer.

Figure 2 illustrates the model architecture. Since our dataset is generated with Pink Trombone, its limited presence in real-world speech may hinder generalization. To address this, we adopt HuBERT as the core for audio feature extraction to enhance robustness.

HuBERT outputs feature vectors of shape $(time_dimension, 1024)$. Because our dataset mainly consists of static short audio segments, the temporal variation within these sequences is limited. To reformulate the problem as a sequence-to-sequence mapping over articulatory spatial positions, we designed a VGG-inspired module that not only compresses the time dimension to 1 but also extracts higher-level acoustic representations across the HuBERT feature space. The resulting $(128, 128)$ representation summarizes the spectral content of the

input and produces a fixed-length feature sequence along the embedding dimension. This feature sequence aligns with the spatial dependencies of the 44 articulatory diameter values and enables the model to learn their structured relationships effectively. The detailed data flow is shown in Fig. 3.

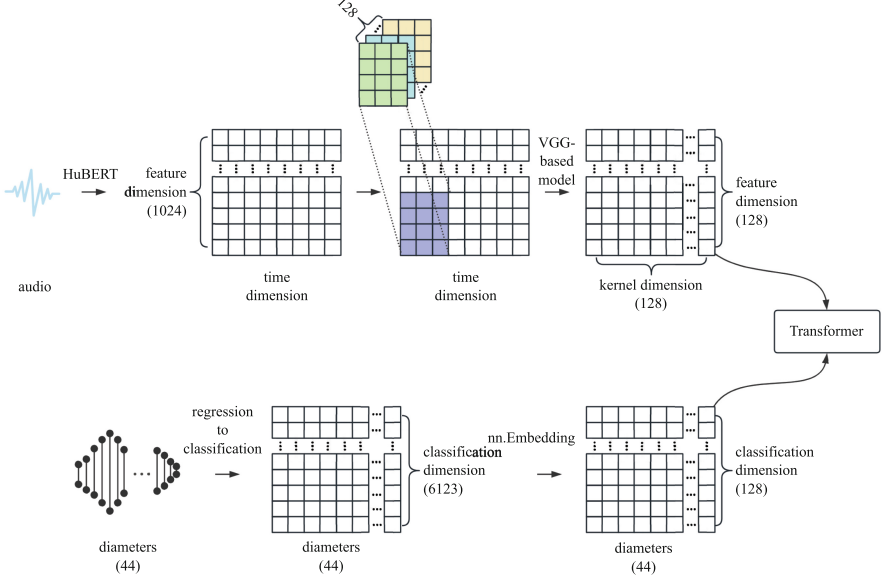


Fig. 3. Detailed data flow of PinkVocalTransformer.

During decoding, the model predicts the articulatory diameter sequence autoregressively. At each prediction step, the decoder is initialized only with a start-of-sequence token and the encoded acoustic features, without access to any ground-truth articulatory values. This prevents any information leakage between the target outputs and ensures that predictions rely solely on the learned dependencies across spatial positions. Causal masking is applied within the decoder so that each predicted position depends only on preceding predictions and not on future targets.

We find that intermediate-layer features outperform the final output for AAI tasks [31–33], offering a better balance between detail and abstraction. These features preserve acoustic and temporal cues essential for modeling articulatory motion.

4 Results

This section presents the experimental results for PinkVocalTransformer on both PT and non-PT tasks. For PT evaluation, we summarize training and validation performance. Because the training data consisted solely of static audio

segments, the evaluation of non-PT audio focuses on the model’s ability to reconstruct vowel-to-vowel (VV) transitions. To assess perceptual quality, we employed ViSQOL [34] and conducted a subjective listening test. For these evaluations, each dynamic utterance in the real recordings was segmented into short frames treated as independent static inputs. The predicted articulatory sequences were then concatenated and smoothed to approximate continuous motion without requiring fully dynamic ground-truth labels.

4.1 Model Training Results

The PinkVocalTransformer was trained using the *pt_data_exlarge* dataset, which includes additive noise to improve robustness. The dataset was split into 80% training and 20% validation sets. We used the AdamW optimizer (initial learning rate of 1×10^{-4} , weight decay of 1×10^{-2}) together with a cosine annealing warm restarts scheduler ($T_0 = 10$ epochs, $T_{\text{mult}} = 1$) and early stopping (patience of 10 epochs, monitoring validation loss) to mitigate overfitting. Training was performed with a batch size of 128 and data shuffling at each epoch. The classification task was optimized using cross-entropy loss and evaluated in terms of accuracy, precision, and recall.

To recover regression targets, classification outputs were mapped to continuous diameter values and evaluated using MSE. The loss converged smoothly, indicating high training stability and effective recovery of articulatory parameters. The best model achieved an accuracy of 0.963, precision of 0.947, and recall of 0.944 on the validation set, indicating strong performance in the PT classification task.

4.2 ViSQOL Evaluation

Because our training used only synthetic audio from Pink Trombone, we needed to evaluate the generalization to real recordings. Since no ground-truth articulatory parameters exist for real speech, we adopted an indirect strategy: if the model produces reasonable parameters, the audio regenerated via Pink Trombone should approximate the original content. We therefore used ViSQOL in speech mode, which focuses on intelligibility and clarity, to compare our model and baselines under realistic conditions. ViSQOL outputs a score between 1 and 5, where higher values indicate greater perceptual similarity to the reference.

We evaluated our model using 24 real human audio samples from a prior AAI study [11], including 11 single-vowel, 7 slow vowel-to-vowel, and 6 complex vowel-dominant samples. The dataset includes 18 male and 6 female samples. All audio was regenerated using multiple baseline methods for comparison. Specifically, we used the outputs of the two-heads decoding VAE proposed in study [11], as well as variants employing Encodec and wav2vec2.0 embeddings as latent representations. Each baseline was tested under *fast* and *slow* configurations reflecting different levels of articulatory dynamics in the training data. We also evaluated optimization-based AAI methods from study [10], but their ViSQOL scores clustered near 2.0, so for clarity we excluded them from the figure.

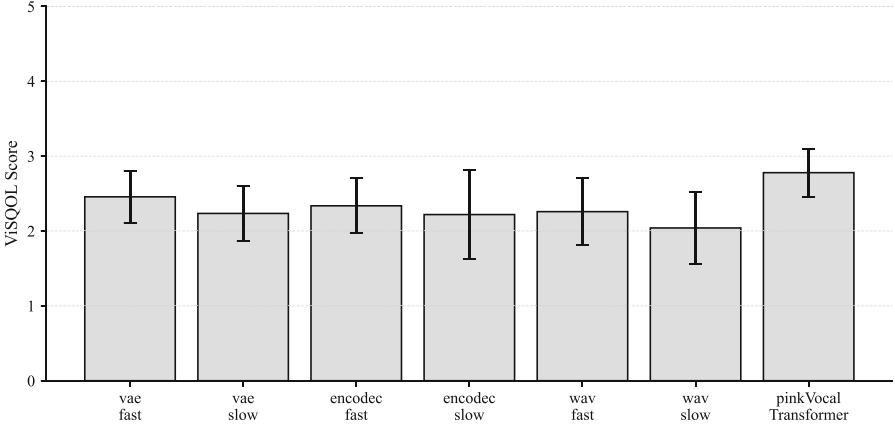


Fig. 4. Mean ViSQOL scores with standard deviations for PinkVocalTransformer and baseline models.

As shown in Fig. 4, PinkVocalTransformer outperformed all baselines in ViSQOL scores. Our model achieved the highest average score, with smaller variability across samples compared to most baselines. Although the absolute scores may appear modest, this is expected since our approach targets acoustic-to-articulatory inversion rather than direct waveform synthesis. The regenerated audio is produced solely for evaluation by feeding predicted articulatory parameters into Pink Trombone, which inevitably introduces timbral and speaker-dependent differences relative to the original recordings. These differences can reduce ViSQOL scores even when the articulatory reconstruction is accurate. While other methods showed lower means and larger error bars, our results remained consistently higher and more stable. Selected audio examples are accessible via our GitHub repository [35].

4.3 Listening Test

To further validate the ViSQOL results, we conducted a single ABX discrimination test comparing our model with the strongest baseline *vae_fast* identified in the objective evaluation. In this test, participants were presented with a reference recording alongside two synthesized candidates and were asked to indicate which synthesized candidate more closely resembled the reference recording in terms of perceived similarity. This procedure captures perceptual similarity rather than overall audio quality. Ten representative samples were selected from the same set of 24 real audio recordings used in the ViSQOL evaluation, ensuring both phonetic diversity and manageable listener effort. In each trial, participants listened to a reference and two synthesized versions, and indicated which one more closely resembled the original. A “neither” option was included to avoid forcing decisions when no clear match was perceived. A total of 21 listeners participated in the test.

Figure 5 presents the results for each listener (L1–L21), showing the number of samples in which they selected the proposed model, the baseline, or neither. Most listeners preferred the audio generated by PinkVocalTransformer, with relatively few neutral responses. For quantitative analysis, we computed ABX accuracy as the proportion of valid trials in which the proposed model was judged closer to the reference. Trials where listeners selected “neither” were excluded from this calculation, as they do not reflect a clear perceptual preference. Based on this criterion, the accuracy reached 81.09%, reinforcing the perceptual advantage of PinkVocalTransformer over the baseline.

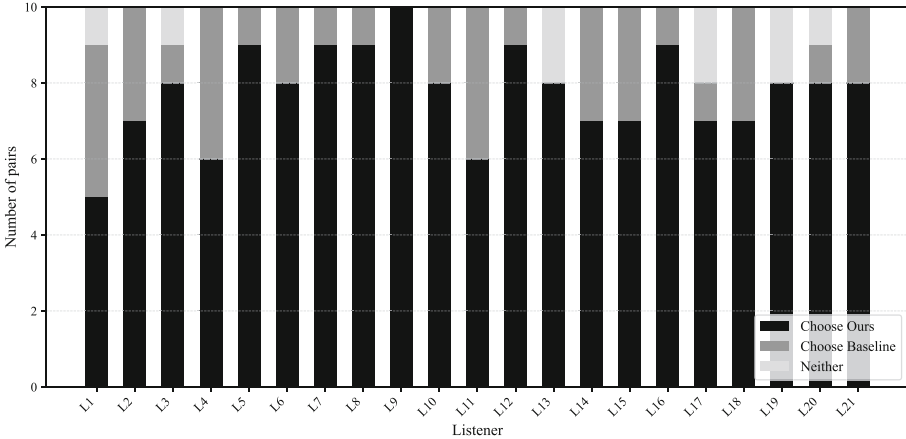


Fig. 5. Listener-level preference distribution in the ABX test.

While the model performed well on most samples, one case showed reduced accuracy. It involved a male-spoken /i/ vowel with dominant low-frequency and stable high-frequency energy. The model emphasized low-frequency cues, causing high-frequency details to be underrepresented and resulting in a dull perceptual quality. In contrast, all other test samples, including those containing /i/ under less extreme conditions, were reconstructed with high perceptual accuracy.

5 Discussion and Conclusion

PinkVocalTransformer shows strong performance in vowel reconstruction and offers a more interpretable formulation of the AAI task. Unlike optimization-based methods that estimate control parameters iteratively, our model is trained once and reused efficiently. In contrast to prior neural approaches that use PT’s user-defined interaction parameters, we directly model vocal tract shape using 44 diameters, which we treat as a continuous spatial sequence rather than independent variables. This formulation enables the decoder to autoregressively predict articulatory configurations, providing a physically grounded and spatially explicit articulation model.

However, the model inherits limitations from its training setup. Trained solely on static, short-duration audio, it performs well on vowels and simple consonants but struggles to reconstruct plosives. To address this, incorporating dynamic yet brief audio and modifying the architecture to model temporal variation may help capture articulatory transitions without increasing overall complexity.

Additionally, the reliance on HuBERT embeddings introduces potential variability when applied to real recordings. Although HuBERT improves robustness compared to conventional acoustic features, it does not explicitly disentangle speaker identity from phonetic content. As a result, predictions may be partially influenced by speaker-dependent cues. Exploring more speaker-invariant representations, such as ContentVec, could help mitigate this effect and improve consistency across diverse speakers.

Another constraint arises from the synthesizer itself. Pink Trombone, being a two-dimensional articulatory model, lacks the natural acoustic richness of real vocal tracts. While this framework offers precise articulatory control, it limits the realism of generated audio.

Future work should address these challenges by exploring alternative articulatory parameterizations and more advanced synthesis techniques. Improving generalization across models and signal domains will be key to developing scalable, black-box AAI systems that remain robust and interpretable across diverse audio conditions.

Acknowledgments. The authors thank Prof. Shalom Lappin for his encouragement and valuable insights.

Disclosure of Interests. The authors have no competing interests to declare.

References

1. Richmond, K.: Estimating articulatory parameters from the acoustic speech signal. Annexe Thesis Digitisation Project 2017 Block 11 (2002)
2. Tokuda, K., et al.: Speech synthesis based on hidden Markov models. *Proc. IEEE* **101**(5), 1234–1252 (2013). <https://doi.org/10.1109/JPROC.2013.2251852>
3. Van den Oord, A., et al.: WaveNet: a generative model for raw audio. In: *Proceedings of the 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9)*, p. 125 (2016)
4. Hunt, A.J., Black, A.W.: Unit selection in a concatenative speech synthesis system using a large speech database. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Atlanta, GA, USA, vol. 1, pp. 373–376 (1996). <https://doi.org/10.1109/ICASSP.1996.541110>
5. Dutoit, T., et al.: The MBROLA project: towards a set of high quality speech synthesizers free of use for non-commercial purposes. In: *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, Philadelphia, PA, USA, vol. 3, pp. 1393–1396 (1996). <https://doi.org/10.1109/ICSLP.1996.607874>
6. Fant, G., Liljencrants, J., Lin, Q.: A four-parameter model of glottal flow. *STL-QPSR*, vol. 4, no. 1985, pp. 1–13 (1985)

7. Fant, G.: Acoustic Theory of Speech Production. The Hague. Mouton, The Netherlands (1960)
8. Thapen, N.: Pink Trombone. <https://dood.al/pinktrombone/>. Accessed 04 July 2025
9. Birkholz, P.: Modeling consonant-vowel coarticulation for articulatory speech synthesis. *PLoS ONE* **8**(4), e60603 (2013). <https://doi.org/10.1371/journal.pone.0060603>
10. Cámara, M., et al.: Optimization techniques for a physical model of human vocalisation. In: 26th International Conference on Digital Audio Effects (DAFx), Copenhagen, Denmark, 4–7 September 2023
11. Cámara, M., et al.: Decoding vocal articulations from acoustic latent representations. In: Proceedings of the AES Europe Convention, Madrid, Spain (2024)
12. Mathur, S., Story, B., Rodriguez, J.: Vocal-tract modeling: Fractional elongation of segment lengths in a waveguide model with half-sample delays. *IEEE Trans. Audio Speech Lang. Process.* **14**, 1754–1762 (2006)
13. Hsu, W.-N., et al.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 3451–3460 (2021). <https://doi.org/10.1109/TASLP.2021.3122291>
14. Dudley, H., Tarnoczy, T.H.: The speaking machine of Wolfgang von Kempelen. *J. Acoust. Soc. Am.* **22**(2), 151–166 (1950). <https://doi.org/10.1121/1.1906583>
15. Kelly, K.L., Lochbaum, C.C.: Speech synthesis. In: Proceedings of the Fourth ICA (1962)
16. Story, B.H.: A parametric model of the vocal tract area function for vowel and consonant simulation. *J. Acoust. Soc. Am.* **117**(5), 3231–3254 (2005)
17. Chenoukh, S., et al.: Voice mimic system using an articulatory codebook for estimation of vocal tract shape. In: Proceedings of the EuroSpeech-97, Rhodes, pp. 429–432 (1997)
18. Ouni, S., Laprie, Y.: Modeling the articulatory space using a hypercube codebook for acoustic-to-articulatory inversion. *J. Acoust. Soc. Am.* **118**(1), 444–460 (2005). <https://doi.org/10.1121/1.1921448>
19. Sorokin, V.N., Leonov, A.S., Trushkin, A.V.: Estimation of stability and accuracy of inverse problem solution for the vocal tract. *Speech Commun.* **30**(1), 55–74 (2000)
20. Saha, P., et al.: Learning joint articulatory-acoustic representations with normalizing flows. In: Proceedings of the Interspeech (2020)
21. Pasad, A., Shi, B., Livescu, K.: Comparative layer-wise analysis of self-supervised speech models. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2023, Rhodes Island, Greece, pp. 1–5 (2023). <https://doi.org/10.1109/ICASSP49357.2023.10096149>
22. Lu, H.-L., Smith, J.O.: Glottal source modeling for singing voice synthesis. In: Proceedings of the ICMC (2000)
23. Fant, G.: The LF-model revisited: transformations and frequency domain analysis. *STL-QPSR*, vol. 2, no. 3 (1995)
24. Südholt, D., et al.: Vocal tract estimation by gradient descent. In: 26th International Conference on Digital Audio Effects (DAFx), Copenhagen, Denmark, 4–7 September 2023
25. Kim, J.W., et al.: Crepe: a convolutional representation for pitch estimation. In: Proceedings of the ICASSP, pp. 161–165 (2018). <https://doi.org/10.1109/ICASSP.2018.8461329>
26. Iman, R.L., Davenport, J.M., Zeigler, D.K.: Latin hypercube sampling (program user’s guide) (1980)

27. Nath, S., Khadilkar, H., Bhattacharyya, P.: Transformers are expressive, but are they expressive enough for regression? arXiv preprint [arXiv:2402.15478](https://arxiv.org/abs/2402.15478) (2024)
28. Freedman, D., Diaconis, P.: On the histogram as a density estimator: L2 theory. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete* **57**(4), 453–476 (1981). <https://doi.org/10.1007/BF01025868>
29. Xu, P., Zhu, X., Clifton, D.A.: Multimodal learning with transformers: a survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(10), 12113–12132 (2023). <https://doi.org/10.1109/TPAMI.2023.3275158>
30. Baevski, A., et al.: Wav2vec 2.0: a framework for self-supervised learning of speech representations. In: *Proceedings of the 34th International Conference on Neural Information Processing Systems (NeurIPS)*, Red Hook, NY, USA, Article no. 1044, pp. 12449–12460. Curran Associates Inc. (2020)
31. Chang, H.-J., Yang, S., Lee, H.-Y.: DistilHuBERT: speech representation learning by layer-wise distillation of hidden-unit BERT. In: *Proceedings of the ICASSP*, pp. 7087–7091 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9747490>
32. Kumar, P., Sukhadia, V.N., Umesh, S.: Investigation of robustness of HuBERT features from different layers to domain, accent and language variations. In: *Proceedings of the ICASSP*, pp. 6887–6891 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9746250>
33. Yoon, J.W., Woo, B.J., Kim, N.S.: HuBERT-EE: early exiting HuBERT for efficient speech recognition. In: *Proceedings of the Interspeech*, pp. 2400–2404 (2024). <https://doi.org/10.21437/Interspeech.2024-80>
34. Chinen, M., et al.: ViSQOL v3: an open source production ready objective speech and audio metric. In: *2020 QoMEX*, pp. 1–6. IEEE (2020)
35. Xu, Z.: PinkVocalTransformer Project Page. <https://zhiyuanxu27.github.io/pinkVocalTransformer/>. Accessed 04 July 2025



CrossMP-SENet: Transformer-Based Cross-Attention for Joint Magnitude-Phase Speech Enhancement

Alexander Zaburdaev[✉], Denis Ivanko[✉], and Dmitry Ryumin[✉]

St. Petersburg Institute for Informatics and Automation of the Russian Academy of Sciences, St. Petersburg Federal Research Center of the Russian Academy of Sciences (SPC RAS), St. Petersburg, Russia
szaburdaev@me.com, {ivanko.d, ryumin.d}@iiias.spb.ru

Abstract. We propose CrossMP-SENet, a speech enhancement architecture that jointly models magnitude and phase spectra using parallel decoding branches connected via cross-attention. Unlike prior approaches that largely treat magnitude and phase enhancement separately or asymmetrically, our method introduces a dedicated cross-attention block that enables deep, bidirectional interaction between the two domains. This design allows the model to leverage complementary spectral cues more effectively during denoising. We adopt a compressed mask prediction framework for magnitude, paired with a dedicated phase decoder, and design a specialized cross-attention mechanism that facilitates information exchange between these representations. To further improve perceptual quality, we incorporate Perceptual Contrast Stretching (PCS). Our experiments on the VoiceBank + DEMAND corpus show that CrossMP-SENet achieves strong performance with a PESQ score of 3.65 using only 2.64 million parameters, outperforming state-of-the-art models with larger architectures. Additionally, we evaluate Transformer and Mamba-based variants and discover that, despite their recent popularity, Mamba blocks do not consistently surpass Transformer-based designs in this context. All models and code are publicly available at <https://github.com/StrangeAlex/CrossMP-SENet>, fostering reproducibility and further research.

Keywords: Speech enhancement · Parallel denoising · Cross-attention · Magnitude-phase spectra · TF-transformers · TF-mamba

1 Introduction

In real-world environments, speech signals are often corrupted by background noise, which degrades the performance of both human auditory perception and automatic speech-based systems such as Automatic Speech Recognition (ASR), speaker verification, and voice communication tools [8, 9, 24]. As a result, Speech

Enhancement (SE) - the task of improving the intelligibility and perceptual quality of speech - has become a central topic in audio signal processing and machine learning research. The goal of SE is to suppress noise from a noisy speech signal and recover a clean, intelligible version of the original speech. Traditional SE approaches were predominantly based on statistical signal processing methods, such as spectral subtraction, Wiener filtering, and minimum mean square error estimators [2, 6]. However, these methods often suffer from limited performance in non-stationary noise environments and may introduce artifacts or distortions [19].

In recent years, the advent of deep learning has revolutionized the field of SE. Deep Neural Networks (DNNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and more recently, transformer-based architectures have demonstrated remarkable capability in modeling complex temporal and spectral structures in noisy speech [10, 23]. Beyond speech processing, deep learning paradigms have also demonstrated transformative impact in a variety of domains [22, 25, 26], further emphasizing their versatility and generalizability. These data-driven methods can learn robust mappings from noisy to clean signals given sufficiently large and diverse training corpora. Despite their progress, most current systems focus heavily on enhancing either the magnitude spectrum or directly estimating the waveform while paying limited attention to the phase information, which is often assumed to be less perceptually relevant or is simply copied from the noisy signal. However, recent studies have shown that accurate phase estimation can contribute significantly to perceptual quality and intelligibility, especially in low Signal-to-Noise Ratio (SNR) conditions [16, 17].

In the Time-Frequency (TF) domain, a speech signal is commonly represented using the Short-Time Fourier Transform (STFT), which decomposes the signal into a complex spectrogram consisting of magnitude and phase components. Traditional DNN-based SE systems often focus on predicting a magnitude mask or estimating a clean magnitude spectrum, while the phase is either ignored or reused from the noisy input [12]. Although this simplification facilitates network training and reduces computational complexity, it imposes an upper bound on enhancement quality due to the phase mismatch problem.

To address this, a growing body of research has explored joint magnitude and phase estimation frameworks. These methods aim to simultaneously improve both spectral components to generate more realistic and artifact-free speech [33]. However, this task is inherently more challenging due to the cyclic nature and higher sensitivity of phase representations. Modeling phase information directly in the STFT domain requires careful architectural design and loss functions to ensure stability and perceptual relevance [35].

In this paper, we propose a codec-style transformer architecture, named CrossMP-SENet, that is designed for joint denoising of both the magnitude and phase spectra of noisy speech signals, and inspired by the original work [16]. Unlike traditional SE models that treat magnitude and phase separately or rely on simplistic phase approximations, the proposed model employs a unified yet dual-branch framework that enables mutual enhancement through cross-

attention mechanisms. Unlike the original work, we incorporate a cross-attention block to better balance the feature representations of magnitude and phase components, which leads to improved SE performance.

The encoder component of our model comprises convolutional and dilated DenseNet blocks [20], which compress the input into a compact latent representation. This representation is then passed through a stack of shared Time-Frequency Transformer (TF-Transformer) layers [15]. These layers are designed to model long-range temporal and spectral dependencies through self-attention in both dimensions. Following this shared stage, the network branches into two independent pathways: one for magnitude enhancement and the other for phase estimation. Each branch consists of dedicated stacks of transformer layers tailored to their respective domains.

A key component of our architecture is the cross-attention module [3, 29], which is introduced after the branch-specific transformers. This module facilitates the feature interaction between magnitude and phase representations, enabling each branch to refine its output by leveraging complementary information from the other. Finally, the model outputs a denoised magnitude spectrum and a reconstructed phase estimate, which are combined via the Inverse Short-Time Fourier Transform (ISTFT) to reconstruct the enhanced waveform.

The remainder of this paper is structured as follows: in Sect. 2, we provide a review of approaches to SE, with a focus on State-of-the-Art (SOTA) results and methodologies on VoiceBank + DEMAND corpus. Section 3 introduces the proposed method in detail. In Sect. 4, we present the experimental setup, including corpus configuration, training details, evaluation metrics, and a comparative analysis with SOTA baselines. Finally, Sect. 5 summarizes our findings and discusses possible directions for future work.

All model architectures, training pipelines, and experiment configurations described in this work are publicly available at GitHub repository: <https://github.com/StrangeAlex/CrossMP-SENet>, to ensure full reproducibility and facilitate further research.

2 Related Work

To ensure a comprehensive and fair evaluation, we select a diverse set of SOTA baselines, including transformer-based, GAN-based, dual-branch, and phase-aware models. These methods have demonstrated strong performance on VoiceBank + DEMAND [4], and we analyze them in greater detail in this section. We choose the VoiceBank [30] + DEMAND [28] corpus for evaluation as it is a standard benchmark in the SE community. It combines high-quality clean speech from the VoiceBank corpus with diverse, real-world noise conditions from the DEMAND corpus, covering a wide range of SNRs. This corpus has been used extensively in recent years, making it easy to compare results across methods.

The VoiceBank + DEMAND corpus has become a central benchmark for evaluating SE methods. It provides noisy-clean speech pairs under various noise conditions, enabling objective comparison via perceptual metrics such as PESQ,

CSIG, CBAK, COVL, and STOI [21]. We selected VoiceBank + DEMAND due to its prevalence in the SE literature, realistic background noise scenarios, and standardized evaluation metrics, enabling direct comparisons with prior work. Many recent methods attempt joint magnitude-phase modeling or efficient transformer-based architectures to improve perceptual quality. Below we summarize key methods.

MP-SENet [16] explicitly denoises magnitude and phase in parallel via a transformer model. Its encoder utilizes convolution-augmented transformers to learn TF dependencies, and its dual-decoder structure refines magnitude via a mask and reconstructs wrapped phase. Trained with multi-level losses and an adversarial metric discriminator, MP-SENet achieves a PESQ of 3.50 on VoiceBank + DEMAND.

PESQeterian [18] is a metric-driven network that directly optimizes the PESQ score using a differentiable approximation of the perceptual metric. Through adversarial training aimed at perceptual quality, it surpasses traditional U-Net models, though achieves moderate improvements compared to more complex spectral-phase models like MP-SENet.

SE-Mamba [5] integrates the Mamba State-Space Model (SSM) into SE and demonstrates strong performance on VoiceBank + DEMAND. Its Perceptual Contrast Stretching (PCS) variant further boosts PESQ to reaching SOTA levels. Mamba-SEUNet-L [32] extends this by embedding Mamba within a U-Net backbone for efficient global modeling, yielding similarly high PESQ (~ 3.59) with low complexity.

Conformer-based Metric GAN (CMGAN) [1] employs TF conformer blocks to jointly estimate magnitude and complex spectrograms, with a metric-driven discriminator enhancing perceptual quality. On VoiceBank + DEMAND, it reaches a PESQ of ~ 3.41 and Segmental Signal-to-Noise Ratio (SSNR) of 11.1dB. Spectral Consistency Preservation (SCP)-CMGAN [36] further extends CMGAN by incorporating SCP to reduce artifacts, leading to improved PESQ and MOS scores.

D²Former [7] is a dual-domain transformer model processing both the time-domain waveform and TF representations. By alternating attention across domains, it achieves improved feature fusion, especially effective in low-SNR conditions, with PESQ ~ 3.48 on VoiceBank + DEMAND.

PCS-CS-WavLM [11] combines PCS with feature representations derived from WavLM- a pre-trained self-supervised speech model. PCS enhances spectral contrast for mask prediction, and WavLM features inject rich contextual embeddings. This hybrid model produces more perceptually natural and intelligible speech by bridging contrastive spectral refinement with powerful latent features.

DeepFilterNet3 [27] builds upon the DeepFilterNet architecture, which was originally designed for two-stage spectral filtering and enhancement. The third iteration refines this structure by integrating denoising blocks with improved spectral filtering capabilities. Evaluated on VoiceBank + DEMAND, DeepFilter-

Net3 maintains strong intelligibility achieves competitive PESQ, while keeping model size modest and enabling real-time execution.

FSPEN [34] is an ultra-lightweight SE network designed for real-time on-device applications. It employs separate full-band and sub-band encoders to extract global and local spectral features, respectively, and introduces an inter-frame path extension mechanism to enhance modeling capacity while preserving computational efficiency. With only $\sim 79\text{K}$ parameters, FSPEN achieves a PESQ score of 2.97 and STOI of 0.942 on VoiceBank + DEMAND, making it highly suitable for deployment in resource-constrained devices.

xLSTM-SENet [13] pioneers the use of extended xLSTM blocks in TF domain SE. Its encoder-decoder framework employs dual decoders and leverages xLSTM layers to capture long-range dependencies efficiently. On VoiceBank + DEMAND, xLSTM-SENet2 matches or surpasses major transformer- and Conformer-based models, demonstrating linear scalability.

PrimeK-Net [14] implements multi-scale spectral processing using Group Prime Kernel Feedforward Channel Attention (GPFCA). It utilizes Deep Separable Dilated Dense Blocks (DSDDb) for efficient encoding and decoding, combined with group prime kernel feedforward channel attention to capture long-, medium-, and short-range spectral dependencies without the periodic overlap issues.

Our proposed cross-attentive dual-branch transformer addresses key limitations: (1) explicit magnitude-phase modeling like MP-SENet and CMGAN, but with deeper branch specialization; (2) global contextual modeling via TF-Transformers, similar to conformer-based methods; (3) cross-attention between branches to refine interdependent spectral features; (4) competitive performance with perceptually motivated techniques like PCS.

We build on the strengths of each paradigm with parallel magnitude-phase refinement, global attention, perceptual optimization and unify them in a coherent codec-style transformer with branch-specific decoders and cross-modal interaction, aiming to set a new standard on the VoiceBank + DEMAND benchmark.

3 Methodology

Our method adopts a codec-style architecture to perform joint denoising of magnitude and phase spectra (see Fig. 1). Given a noisy waveform $y \in \mathbb{R}^L$, we extract its TF representation using the STFT, which results in spectra of magnitude $Y_m \in \mathbb{R}^{T \times F}$ and wrapped phase $Y_p \in \mathbb{R}^{T \times F}$. These are combined into a two-channel input feature $Y_{\text{in}} \in \mathbb{R}^{T \times F \times 2}$ after applying a power-law compression to the magnitude spectrum for better mask predictability.

The encoder first transforms the input into a compressed TF representation using convolutional and dilated DenseNet blocks [20]. This representation is then passed through a stack of N shared TF-Transformers layers to capture global contextual dependencies. The output is subsequently split into two branches: one for magnitude and one for phase. Each branch processes the shared representation independently via M and P TF-Transformer layers, respectively.

A dedicated cross-attention module then facilitates interaction between magnitude and phase representations, allowing each branch to refine its features based on complementary spectral information. Finally, the magnitude mask decoder estimates a clean magnitude spectrum, while the phase decoder reconstructs the wrapped phase. These outputs are used together in ISTFT to create the enhanced waveform \hat{x} .

Branch separation aims to better capture the distinct characteristics of magnitude and phase by allowing them to be processed in separate transformer modules. While the shared TF-Transformer extract general features from the input, the branched transformers specialize in domain-specific refinement. The cross-attention block then facilitates information exchange, enabling the magnitude and phase representations to benefit from each other even more.

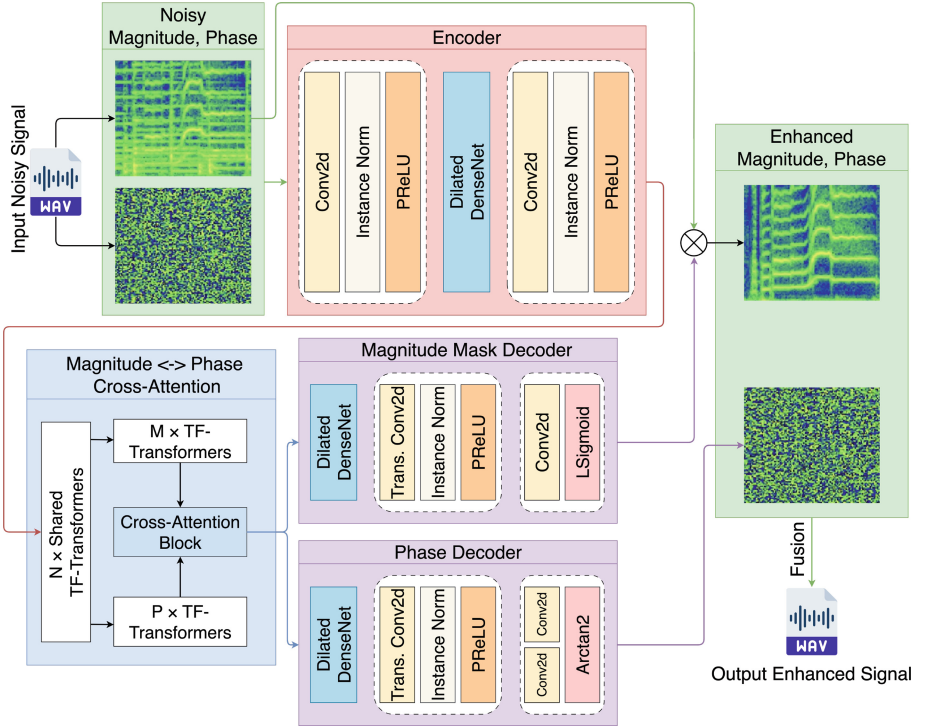


Fig. 1. Overall structure of the CrossMP-SENet with cross-attention mechanism.

3.1 Model Architecture

Encoder. The encoder projects the input into a compact latent representation using three components: an initial convolutional block, a dilated DenseNet,

and a second convolutional block. Each convolutional block includes a 2D convolution layer, instance normalization, and PReLU activation. The first block increases feature dimensionality, while the second reduces temporal resolution. The DenseNet applies dilated convolutions with rates $\{1, 2, 4, 8\}$ and uses dense connections to improve gradient flow and extend receptive fields.

Cross-Attention Module. After the branch-specific TF-Transformer layers for magnitude and phase, our architecture integrates a dedicated cross-attention module (see Fig. 1). This component facilitates bidirectional information exchange between the magnitude and phase representations, enabling each decoder branch to refine its output by leveraging complementary spectral features from the other domain.

Technically, the cross-attention is implemented using a standard query-key-value attention mechanism, where the feature maps from one branch serve as queries, while the corresponding feature maps from the other branch act as keys and values. This design allows each branch to dynamically adjust its feature representations based on the contextual information extracted from the other spectral domain.

The inclusion of the cross-attention module addresses one of the key limitations of prior magnitude-phase joint models - the lack of inter-domain dependency modeling. By explicitly promoting interaction between magnitude and phase representations at the decoding stage, the proposed module helps mitigate inter-spectral artifacts and improves the overall perceptual quality of the enhanced signal.

Magnitude Mask Decoder. To estimate the clean magnitude spectrum from a noisy input, our model predicts a multiplicative mask that adjusts the noisy magnitude towards a cleaner version. However, directly estimating this mask can be unstable because its values are not naturally bounded. To address this, we apply a power compression technique, which reduces the dynamic range of the mask values, making it easier for the model to learn. Specifically, we compress the magnitude spectrum using a power-law function $Y_m^c = Y_m^{0.3}$, which flattens high-energy components for easier mask prediction, and the model is trained to predict this compressed version instead.

To produce the compressed mask, we use a customized sigmoid function that is parameterized and learnable. This function allows the model to control the range and steepness of the output dynamically across frequency bins, improving flexibility and accuracy during training.

The decoder responsible for producing the clean magnitude consists of several components: a dilated DenseNet to expand the receptive field, a transposed convolution layer to increase the resolution, and a final convolution followed by the learnable sigmoid activation. Once the compressed mask is predicted, it is multiplied element-wise with the compressed noisy magnitude spectrum. Finally, we reverse the compression by raising the result to the power of the inverse exponent, effectively recovering the enhanced magnitude spectrum.

Phase Decoder. The phase decoder is designed to directly estimate the wrapped phase of the clean signal. To do this, it first processes the input using a dilated DenseNet to capture long-range dependencies in the TF domain, followed by a deconvolutional layer to upsample the features back to the original resolution.

Next, the decoder uses two parallel convolutional layers to predict two intermediate feature maps, which can be interpreted as pseudo-real and pseudo-imaginary components of the phase. These components are not actual real and imaginary parts of a complex number, but are treated similarly to allow a smooth and stable phase reconstruction.

To derive the final phase estimate from these components, a specially adjusted arctangent function is applied. This modification ensures that the resulting phase values are correctly wrapped within the standard range and avoids common discontinuities or ambiguities that arise with typical phase calculations. The use of sign-based adjustments ensures consistent phase prediction across all quadrants of the complex plane.

3.2 Training Objectives

The model is trained using multiple complementary losses. First, a time-domain loss compares the enhanced waveform directly with the clean reference signal. This encourages the model to produce outputs that are close to the target audio in the time domain.

Time-domain loss:

$$\mathcal{L}_{\text{time}} = \mathbb{E}_{x, \hat{x}} [\|x - \hat{x}\|_1] \quad (1)$$

Next, a magnitude loss is used to ensure that the predicted magnitude spectrum closely matches the true clean magnitude. This is measured using the mean squared error between the two.

Magnitude loss:

$$\mathcal{L}_{\text{mag}} = \mathbb{E}_{X_m, \hat{X}_m} [\|X_m - \hat{X}_m\|_2^2] \quad (2)$$

To account for the full complex spectrum, a complex loss is added. It evaluates the difference between the real and imaginary parts of the predicted and clean spectrograms. This helps the model better reconstruct the overall spectral structure.

Complex loss:

$$\mathcal{L}_{\text{com}} = \|X_r - \hat{X}_r\|_2^2 + \|X_i - \hat{X}_i\|_2^2 \quad (3)$$

To improve the perceptual quality of the enhanced audio, the model incorporates an adversarial component. A discriminator network is trained to predict the PESQ score for the enhanced audio. The generator then learns to optimize its output to achieve a higher PESQ, encouraging more natural-sounding results.

For phase estimation, additional calculation is taken due to the wrapping nature of phase values. A specialized anti-wrapping function is used to measure differences between predicted and true phase values in a way that accounts for circularity. E.g., the fact that 0 and 2π represent the same angle.

Three types of phase-related losses are defined using this wrapping-aware approach: instantaneous phase loss, which measures the direct difference between predicted and true phase values. Group delay loss, which evaluates how phase changes across frequency, reflecting the timing of different frequency components. Instantaneous angular frequency loss, which considers how phase evolves over time, capturing temporal dynamics in the signal. These three components are combined into a total phase loss, ensuring robust phase reconstruction.

Finally, the overall objective function for the generator is a weighted sum of all the above losses. The weights for each component are carefully chosen based on empirical performance to balance their influence during training, forming the following loss function:

$$\mathcal{L}_G = 0.2\mathcal{L}_{\text{time}} + 0.9\mathcal{L}_{\text{mag}} + 0.1\mathcal{L}_{\text{com}} + 0.05\mathcal{L}_{\text{metric}} + 0.3\mathcal{L}_{\text{phase}} \quad (4)$$

This combined loss encourages the model to produce outputs that are not only mathematically accurate but also perceptually relevant.

4 Experiments

4.1 Corpus and Evaluation Protocol

All experiments were conducted using the publicly available VoiceBank + DEMAND corpus. The training set contains 11,572 utterances from 28 speakers, while the test set comprises 824 utterances from 2 unseen speakers. No separate validation set was used. All waveforms were downsampled to 16 kHz. The noisy samples were created by mixing clean speech with various types of real-world noise at different SNR levels, as defined in the corpus.

We use PESQ and STOI metrics to assess speech quality and intelligibility. In our experiments, all models achieve a STOI score of 96 (rounded to an integer percent), indicating near-ceiling intelligibility performance on this corpus. Therefore, we omit STOI from some tables and focus primarily on PESQ and perceptual metrics like CSIG, CBAK, COVL, and SSNR for analysis.

4.2 Ablation Study

To investigate the contribution of different model architectures, we conducted a series of experiments, as shown in Table 1. In addition to the main Transformer-based architecture, we also conducted experiments using Mamba blocks as a replacement for the TF-Transformer layers, motivated by their recent popularity and promising performance in sequence modeling tasks. However, in our experiments, the Mamba-based models did not outperform the Transformer-based ones in terms of SE quality, and thus we retained Transformers in our final design.

In particular, we explored the impact of replacing magnitude TF-Transformers with Mamba blocks (“Mamba/TF separate”), adding PCS, reducing model size, and removing Mamba blocks entirely.

Table 1. Results for different model variants on the VoiceBank+DEMAND test set. Bold indicates the best value in each column

Architecture	PESQ	Params	CSIG	CBAK	COVL	SSNR
Mamba/TF separate (large)	3.44	5.95M	4.73	3.93	4.20	10.67
Mamba/TF separate (large) + PCS	3.64	5.95M	4.78	3.60	4.32	4.17
TF only	3.48	2.64M	4.72	3.94	4.22	10.50
TF only + PCS	3.65	2.64M	4.76	3.61	4.33	4.13
Mamba/TF separate (small)	3.35	2M	4.66	3.87	4.11	10.47
Mamba/TF separate (small) + PCS	3.56	2.26M	4.71	3.57	4.25	4.15

To improve generalization and perceptual quality, we introduced PCS. Adding PCS to the large Mamba/TF improved PESQ from 3.44 to 3.64 and COVL from 4.20 to 4.32, while also yielding the best CSIG score (4.78). Notably, the Transformer-only variant without Mamba blocks achieved strong performance with only 2.64M parameters. Adding PCS to this variant further improved PESQ to 3.65 and COVL to 4.33, achieving the best scores in both metrics. This indicates that Transformer-based representations remain highly effective, and that PCS plays a crucial role in improving perceptual quality.

Smaller models using Mamba/TF showed acceptable performance with a reduced parameter footprint (as low as 2M), but their overall enhancement quality lagged behind larger configurations. The PCS addition still provided consistent gains in PESQ and COVL for the small variants, confirming the robustness of this training strategy.

Overall, while Mamba blocks have gained attention for efficient sequence modeling, in our experiments they did not consistently outperform the Transformer-based architecture in terms of perceptual enhancement metrics. Therefore, we retained the Transformer-based design for our final model.

4.3 Comparison with Advanced SE Methods

Table 2 summarizes the performance of our proposed model, CrossMP-SENet, in comparison to several recent SE architectures. We focus on two primary metrics: PESQ, which evaluates perceptual speech quality, and model size in millions of parameters. All models in the comparison achieve a STOI score of 96 on the VoiceBank + DEMAND test set, ensuring comparability in terms of intelligibility.

Table 2. Comparison with other models based on PESQ and model size (in millions of parameters). All models achieve STOI = 96 and are evaluated on the VoiceBank+DEMAND test set

Architecture	PESQ	Params
Mamba-SEUNet [32]	3.73	6.28M
SEMamba [5]	3.69	2.25M
CrossMP-SENet (ours)	3.65	2.64M
ZipEnhancer [31]	3.63	2.04M
PrimeK-Net [14]	3.61	1.41M
MP-SENet [16]	3.60	2.26M

Our CrossMP-SENet method achieves a PESQ of 3.65 with 2.64 million parameters, demonstrating strong performance while maintaining relatively low model complexity. While Mamba-SEUNet achieves a slightly higher PESQ score of 3.73, it does so with more than twice the parameter count (6.28M). Similarly, SEMamba attains a PESQ of 3.69 with a comparable model size to ours (2.25M), indicating a favorable trade-off.

Despite recent interest in Mamba-based architectures, our experiments suggest that simply replacing temporal fusion modules in MP-SENet with Mamba blocks (as done in SEMamba and Mamba-SEUNet) does not consistently yield superior performance.

Our focus was not merely on chasing incremental PESQ gains but on building an efficient, open, and well-balanced model. To this end, CrossMP-SENet achieves strong performance with just 2.64M parameters—outperforming the original MP-SENet in PESQ and maintaining competitive scores in CSIG, CBAK, and COVL. All our checkpoints and code are released publicly to support reproducibility and further research.

Compared to other SOTA models such as ZipEnhancer (3.63 PESQ, 2.04M params), PrimeK-Net (3.61 PESQ, 1.41M params), and MP-SENet (3.60 PESQ, 2.26M params), our CrossMP-SENet outperforms them in PESQ while remaining lightweight and efficient.

Ultimately, our goal was to improve upon MP-SENet without significantly increasing model complexity. CrossMP-SENet fulfills this goal while offering a competitive alternative to more complex or proprietary solutions

5 Conclusion

In this work, we proposed CrossMP-SENet, a lightweight and effective SE model that leverages cross-attention for parallel magnitude-phase separation. Our experiments on the widely used VoiceBank + DEMAND corpus demonstrate that CrossMP-SENet achieves competitive or superior performance compared to recent SOTA models, with a strong PESQ score of 3.65 and 2.64 million

parameters. All models in our researches reached the STOI intelligibility ceiling of 96%, allowing our evaluation to focus on perceptual metrics.

We conducted extensive ablation studies to analyze the impact of model variants, including the use of recently popular Mamba blocks as a replacement for Transformer layers. While Mamba-based models showed acceptable performance, they did not consistently outperform the Transformer-based design in our SE setting. As a result, we retained the Transformer structure for our final model. Additionally, we found that integrating PCS consistently improved perceptual metrics, particularly PESQ and COVL, across all model configurations.

Our findings highlight the effectiveness of cross-attention-driven magnitude-phase fusion, and reaffirm the robustness of Transformer-based architectures. All code and models are publicly available at: <https://github.com/StrangeAlex/CrossMP-SENet>, to assure reproducibility and to promote further research.

Our current architecture employs a shared encoder followed by branch-specific decoding with cross-attention, future work could explore the use of separate encoders for magnitude and phase representations. Introducing early separation of spectral domains may allow the model to learn more specialized feature hierarchies, and integrating cross-attention between these encoders could enable richer inter-domain interactions from the initial stages of processing. Additionally, experimenting with hierarchical or multi-scale cross-attention mechanisms may further improve the fidelity of phase-aware enhancement.

Acknowledgments. This work was financially supported by the State Research project No. FFZF-2025-0003.

References

1. Abdulatif, S., Cao, R., Yang, B.: CMGAN: conformer-based metric-GAN for monaural speech enhancement. *IEEE/ACM Trans. Audio Speech Lang. Process.* 2477–2493 (2024). <https://doi.org/10.1109/TASLP.2024.3393718>
2. Anees, M.: Speech coding techniques and challenges: a comprehensive literature survey. *Multimed. Tools Appl.* **83**(10), 29859–29879 (2024). <https://doi.org/10.1007/s11042-023-16665-3>
3. Axyonov, A., Ryumin, D., Ivanko, D., Kashevnik, A., Karpov, A.: Audio-visual speech recognition in-the-wild: multi-angle vehicle cabin corpus and attention-based method. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8195–8199. IEEE (2024). <https://doi.org/10.1109/ICASSP48485.2024.10448048>
4. Botinhao, C.V., Wang, X., Takaki, S., Yamagishi, J.: Investigating RNN-based speech enhancement methods for noise-robust text-to-speech. In: *ISCA Speech Synthesis Workshop (SSW)*, pp. 159–165 (2016). <https://doi.org/10.21437/SSW.2016-24>
5. Chao, R., et al.: An investigation of incorporating mamba for speech enhancement. In: *IEEE Spoken Language Technology Workshop (SLTW)*, pp. 302–308 (2024). <https://doi.org/10.1109/SLT61566.2024.10832332>

6. Das, N., Chakraborty, S., Chaki, J., Padhy, N., Dey, N.: Fundamentals, present and future perspectives of speech enhancement. *Int. J. Speech Technol.* **24**(4), 883–901 (2020). <https://doi.org/10.1007/s10772-020-09674-2>
7. He, J., Gao, Y., Zhang, T., Zhang, Z., Wu, F.: D²Former: jointly learning hierarchical detectors and contextual descriptors via agent-based transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2904–2914 (2023). <https://doi.org/10.1109/CVPR52729.2023.00284>
8. Ivanko, D., Ryumin, D., Axyonov, A., Kashevnik, A.: Speaker-dependent visual command recognition in vehicle cabin: methodology and evaluation. In: Karpov, A., Potapova, R. (eds.) *SPECOM 2021. LNCS (LNAI)*, vol. 12997, pp. 291–302. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-87802-3_27
9. Ivanko, D., et al.: DAVIS: driver’s audio-visual speech recognition. In: *INTER-SPEECH*, pp. 1141–1142 (2022)
10. Jannu, C., Vanambathina, S.D.: An overview of speech enhancement based on deep learning techniques. *Int. J. Image Graph.* **25**(01), 2550001 (2025). <https://doi.org/10.1142/S0219467825500019>
11. Khan, M.S., La Quatra, M., Hung, K.H., Fu, S.W., Siniscalchi, S.M., Tsao, Y.: Exploiting consistency-preserving loss and perceptual contrast stretching to boost ssl-based speech enhancement. In: *International Workshop on Multimedia Signal Processing (MMSP)*, pp. 1–6 (2024)
12. Kim, E., Seo, H.: SE-conformer: time-domain speech enhancement using conformer. In: *INTER-SPEECH*, pp. 2736–2740 (2021). <https://doi.org/10.21437/Interspeech.2021-2207>
13. Kühne, N.L., Østergaard, J., Jensen, J., Tan, Z.H.: xLSTM-SENet: xLSTM for single-channel speech enhancement. In: *INTER-SPEECH* (2025)
14. Lin, Z., Wang, J., Li, R., Shen, F., Xuan, X.: PrimeK-net: multi-scale spectral learning via group prime-kernel convolutional neural networks for single channel speech enhancement. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10890034>
15. Liu, J., Li, Z.: TF-transformer: temporal-frequency transformer for OFDM signal recognition. In: *IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–6 (2025). <https://doi.org/10.1109/WCNC61545.2025.10978396>
16. Lu, Y.X., Ai, Y., Ling, Z.H.: MP-SENet: a speech enhancement model with parallel denoising of magnitude and phase spectra. In: *INTER-SPEECH*, pp. 3834–3838 (2023). <https://doi.org/10.21437/Interspeech.2023-1441>
17. Lu, Y.X., Ai, Y., Ling, Z.H.: Explicit estimation of magnitude and phase spectra in parallel for high-quality speech enhancement. *Neural Netw.* 107562 (2025). <https://doi.org/10.1016/j.neunet.2025.107562>
18. de Oliveira, D., Welker, S., Richter, J., Gerkmann, T.: The PESQetarian: on the relevance of goodhart’s law for speech enhancement. In: *INTER-SPEECH*, pp. 3854–3858 (2024). <https://doi.org/10.21437/Interspeech.2024-2051>
19. O’Shaughnessy, D.: Speech enhancement - a review of modern methods. *IEEE Trans. Hum.-Mach. Syst.* **54**(1), 110–120 (2024). <https://doi.org/10.1109/THMS.2023.3339663>
20. Pandey, A., Wang, D.: Densely connected neural network with dilated convolutions for real-time speech enhancement in the time domain. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6629–6633 (2020). <https://doi.org/10.1109/ICASSP40776.2020.9054536>

21. Pirklbauer, J., et al.: Evaluation metrics for generative speech enhancement methods: issues and perspectives. In: Speech Communication; ITG Conference, pp. 265–269. VDE (2023). <https://doi.org/10.30420/456164052>
22. Ryumin, D., Ivanko, D., Axyonov, A.: Cross-language transfer learning using visual information for automatic sign gesture recognition. *Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci.* **48**, 209–216 (2023). <https://doi.org/10.5194/isprs-archives-XLVIII-2-W3-2023-209-2023>
23. Ryumin, D., Axyonov, A., Ryumina, E., Ivanko, D., Kashevnik, A., Karpov, A.: Audio-visual speech recognition based on regulated transformer and spatio-temporal fusion strategy for driver assistive systems. *Expert Syst. Appl.* **252**, 124159 (2024). <https://doi.org/10.1016/j.eswa.2024.124159>
24. Ryumin, D., Ivanko, D., Ryumina, E.: Audio-visual speech and gesture recognition by sensors of mobile devices. *Sensors* **23**(4), 2284 (2023). <https://doi.org/10.3390/s23042284>
25. Ryumina, E., Markitantov, M., Ryumin, D., Karpov, A.: Ocean-AI framework with emoformer cross-hemiface attention approach for personality traits assessment. *Expert Syst. Appl.* **239**, 122441 (2024). <https://doi.org/10.1016/j.eswa.2023.122441>
26. Ryumina, E., Ryumin, D., Axyonov, A., Ivanko, D., Karpov, A.: Multi-corpus emotion recognition method based on cross-modal gated attention fusion. *Pattern Recogn. Lett.* **190**, 192–200 (2025). <https://doi.org/10.1016/j.patrec.2025.02.024>
27. Schroter, H., Escalante-B, A.N., Rosenkranz, T., Maier, A.: DeepFilterNet: a low complexity speech enhancement framework for full-band audio based on deep filtering. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7407–7411 (2022). <https://doi.org/10.1109/ICASSP43922.2022.9747055>
28. Thiemann, J., Ito, N., Vincent, E.: DEMAND: A Collection of Multi-channel Recordings of Acoustic Noise in Diverse Environments (2013). <https://doi.org/10.5281/zenodo.1227121>
29. Vaswani, A., et al.: Attention is all you need. In: *Advances in Neural Information Processing Systems (NeurIPS)*, pp. 6000–6010 (2017)
30. Veaux, C., Yamagishi, J., King, S.: The voice bank corpus: design, collection and data analysis of a large regional accent speech database. In: *IEEE International Conference Oriental COCOSDA Held Jointly with Conference on Asian Spoken Language Research and Evaluation (O-COCOSDA/CASLRE)*, pp. 1–4 (2013). <https://doi.org/10.1109/ICSODA.2013.6709856>
31. Wang, H., Tian, B.: ZipEnhancer: dual-path down-up sampling-based zipformer for monaural speech enhancement. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10888703>
32. Wang, J., Lin, Z., Wang, T., Ge, M., Wang, L., Dang, J.: Mamba-SEUNet: mamba UNet for monaural speech enhancement. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10889525>
33. Wang, Z.Q., Cornell, S., Choi, S., Lee, Y., Kim, B.Y., Watanabe, S.: TF-GridNet: making time-frequency domain models great again for monaural speaker separation. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5 (2023). <https://doi.org/10.1109/ICASSP49357.2023.10094992>
34. Yang, L., Liu, W., Meng, R., Lee, G., Baek, S., Moon, H.G.: FSPEN: an ultra-lightweight network for real time speech enhancement. In: *IEEE International Con-*

- ference on Acoustics, Speech and Signal Processing (ICASSP), pp. 10671–10675 (2024). <https://doi.org/10.1109/ICASSP48485.2024.10446016>
35. Yin, D., Luo, C., Xiong, Z., Zeng, W.: PHASEN: a phase-and-harmonics-aware speech enhancement network. In: AAAI Conference on Artificial Intelligence, vol. 34, pp. 9458–9465 (2020). <https://doi.org/10.1609/AAAI.V34I05.6489>
 36. Zadorozhnyy, V., Ye, Q., Koishida, K.: SCP-GAN: self-correcting discriminator optimization for training consistency preserving metric GAN on speech enhancement tasks. In: INTERSPEECH, pp. 2463–2467 (2023). <https://doi.org/10.21437/Interspeech.2023-456>



Adaptive Singing Voice Enhancement for Live Stages

Jia-Lien Hsu^(✉)  and Pei-Wen Chien

Department of Computer Science and Information Engineering,
Fu Jen Catholic University, New Taipei 24205, Taiwan, R.O.C.

alien@csie.fju.edu.tw

<https://alienatfju.github.io/lab/>

Abstract. Live concert recordings play a crucial role in the music industry but often suffer from complex audio issues not typically encountered in studio productions. These include microphone phase differences, crosstalk, background noise, signal dropouts, clipping, excessive sibilance, vocal distortion, and performance mistakes. Traditional restoration approaches, such as manual editing or re-recording, are labor-intensive, costly, and often fail to preserve the authenticity of the live performance. This study presents a generative AI-based system for restoring singing voices in live concert recordings. The proposed framework uses cross-modal feature transformation to repair problematic segments by leveraging musical score information. The system takes as input a short vocal audio clip (4–6 bars), a lyrics file, and the corresponding musical score, and outputs a high-quality, restored vocal segment. The architecture integrates three key components: SOFA, a forced-alignment tool for precise synchronization between vocals and lyrics; DiffSinger, a diffusion-based singing voice synthesis model that extracts phoneme-level features from the score; and VoiceCraft, a text-to-speech model adapted as the core generative module due to its strong acoustic preservation capabilities. A central component of the system is a deep learning-based adapter module that enables cross-modal mapping by transforming musical feature representations into VoiceCraft-compatible text embeddings. Audio generation is carried out through an autoregressive Transformer-based architecture with attention masking and Encodec quantization, ensuring temporal consistency and audio quality. The system effectively retains the original singer's timbre and the environmental acoustics of the recording, allowing the restored segments to blend seamlessly with the original performance. This research offers a novel, efficient, and high-fidelity solution for vocal restoration in live music recordings, overcoming the limitations of traditional methods and introducing a new technological direction for the music production industry.

Keywords: Vocal restoration · Concert recording · Generative AI · Cross-modal feature transformation

1 Introduction

Live concerts play an indispensable role in the contemporary music industry. For singers, they serve not only as a stage to showcase their live performance abilities but also as a crucial means of enhancing brand value and market influence. Concerts provide a comprehensive platform to express musical concepts through arrangements, staging, and visual design. More importantly, they offer valuable opportunities for artists to interact directly with fans and foster emotional resonance. As such, concerts are not merely a part of an artist's career journey but a vital form of artistic expression.

In the music industry, live concert recordings are commercially valuable products. They faithfully capture the atmosphere and key moments of live interaction, allowing fans who could not attend in person—or those wishing to relive the experience—to enjoy the performance. When these recordings are transformed into physical or digital media products, they not only generate additional revenue for artists and related industries but also help promote their brand and merchandise. However, audio post-production is essential to ensure that the recordings are presented in their best form.

Compared to studio recordings, live concert recordings involve more complex technical challenges. Though multitrack recordings of vocals and instruments are often used, they are affected by various factors such as venue acoustics, sound reflections, and interference between instruments. Additionally, unpredictable events during the performance can directly impact recording quality. The available materials for post-production are usually limited to recordings from the performance day and rehearsals, making the restoration process significantly more challenging than that of a studio album.

Among the various elements of a live recording, the lead vocal is often the listener's primary focus. As such, vocal processing quality is a crucial and central concern in mixing. However, post-production faces multiple technical issues, including microphone phase differences, audio bleed/spill/leakage, ambient noise, signal dropouts, popping, excessive sibilance, vocal cracking or breaking, and performance errors.

Currently, the industry addresses these issues mainly through two approaches: traditional audio processing tools (multiple plugins within a digital audio workstation), and re-recording when possible. Moreover, additional processing is needed to replicate the live acoustic characteristics, further increasing workload.

Given these challenges, there is significant research value in moving beyond conventional restoration approaches. Rather than focusing solely on correcting flaws in the original audio, generative technologies offer the potential to create new, high-quality audio that preserves the unique characteristics of the original performance—assisting mixing engineers in achieving more efficient and effective post-production.

1.1 Research Objective

This study aims to develop a singing voice restoration system specifically designed for live concert recordings, addressing the limitations of traditional audio restoration methods. The system targets common issues in live audio such as dropouts, vocal cracking, and performance errors, offering an intelligent solution for restoration.

Unlike conventional methods that attempt to patch flaws in the original signal, the proposed system employs generative techniques to synthesize high-quality replacement audio using deep learning. In addition to solving audio issues, the system’s core objective is to preserve the singer’s original vocal timbre and stylistic characteristics while maintaining the acoustic ambiance of the live performance, ensuring a seamless blend between restored and original segments.

As shown in Fig. 1, the system relies on three primary inputs:

- The original vocal audio segment containing the defect.
- The lyrical phrase corresponding to the segment.
- The complete musical score for that song portion.

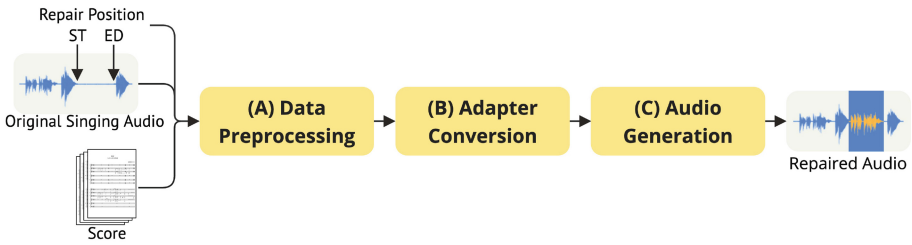


Fig. 1. System architecture and research objective.

To achieve this, the system is composed of three core modules:

- Preprocessing module—Extracts features (lyrics, pitch, pitch duration) from the musical score and aligns them precisely with the original vocal using score-lyric-audio synchronization.
- Adapter module—Transforms musical features into representations understandable by the generative model.
- Audio generation module—Uses the aligned lyrics and score to locate the target segment, then generates a high-quality restoration conditioned on the adapter’s output.

The system is based on VoiceCraft, a TTS model known for its ability to preserve environmental acoustic features. By developing a custom adapter mechanism, this study seeks to extend VoiceCraft’s capabilities beyond text, enabling it to process musical input and generate expressive, contextually integrated singing voice. This represents a novel and efficient solution for live concert vocal restoration in music production.

2 Related Work

The Evolution of TTS Models. The TTS (Text-to-Speech) has evolved from traditional methods such as articulatory, formant, and concatenative synthesis to statistical parametric speech synthesis (SPSS), which introduced machine learning into acoustic modeling and vocoding [21]. The advent of deep learning marked a major shift, with WaveNet [22] enabling direct raw waveform generation and improving speech naturalness.

Later systems transitioned to end-to-end architectures. Tacotron [25] used an encoder-decoder framework to generate mel-spectrograms from text, while FastSpeech [18] adopted a non-autoregressive Transformer for faster inference. FastSpeech 2 [17] removed reliance on teacher models, and FastSpeech 2s enabled direct waveform generation. Parallel Tacotron 2 [4] further improved alignment learning with a fully differentiable duration model, reflecting trends toward higher efficiency, better quality, and full end-to-end synthesis.

Zero-Shot and Few-Shot TTS. With the growing demand for personalized speech synthesis, Zero-Shot and Few-Shot TTS have emerged, aiming to synthesize speaker-specific voices with minimal data. Zero-Shot TTS uses reference audio without labeled data, while Few-Shot TTS leverages limited labeled samples. Key challenges include preserving speaker characteristics under data scarcity.

Attentron [3] enhanced Tacotron 2 with dual encoders and attention for variable-length reference inputs. Meta-TTS [9] applied meta-learning to improve adaptation efficiency. YourTTS [2] and ZMM-TTS [5] extended Zero-Shot TTS to multilingual and low-resource settings using multilingual training and self-supervised discrete representations, respectively.

Jeong et al. [10] unified multilingual and Zero-Shot TTS via a two-stage transfer learning framework. Wang et al. [24] introduced the USAT framework with lightweight adapters for accent and non-native speaker adaptation and proposed the ESLTTS benchmark.

VoiceCraft [16] integrates speech editing with Zero-Shot TTS via token infilling, but lacks modeling of musical context. This study extends VoiceCraft for singing voice restoration in live concerts by incorporating musical scores and adapter modules.

SVS Task. Singing Voice Synthesis (SVS) aims to generate expressive singing from musical scores, facing challenges in pitch, rhythm, and naturalness modeling. Early methods included HMM-based systems [12] that addressed F0 sparsity and alignment issues. Neural approaches such as NPSS [1], XiaoIceSing [15], and ByteSing [6] improved spectral modeling, pitch control, and alignment robustness.

DiffSinger [14] and RMSSinger [7] adopted diffusion models for better quality and realistic score handling. Zero-Shot SVS methods like NaturalSpeech 2 [19] and Wang et al. [23] further advanced cross-style and cross-lingual singing generation using limited data.

Despite these advances, SVS still faces challenges in low-resource scenarios and modeling accuracy for expressive, high-quality singing output.

Cross-Modal Learning and Applications in Music. Cross-modal learning enables understanding and transformation across different data modalities. In the music domain, Li et al. [11] categorized research into music-driven, music-targeted, and bidirectional interactions, emphasizing challenges due to music’s abstract semantics and long-range dependencies.

Several methods have been proposed for cross-modal representation alignment. ICCN [20] models audio-vision interaction using outer-product embeddings. MuLan [8] employs dual encoders for audio-text retrieval. Yu et al. [26] focused on audio-lyrics learning using dual-branch networks.

Adapters are emerging as efficient tools for cross-modal conversion by enabling modular fine-tuning. LAVISH [13] introduced latent-token-based adapters for audio-visual fusion in vision Transformers. However, their application in music-text conversion remains underexplored. This study investigates adapter-based music-to-text transformation, addressing a gap in current cross-modal learning research.

3 Method

3.1 System Architecture

This study proposes a generative AI-based singing voice restoration system designed to repair degraded segments in live concert recordings. The system architecture comprises several specialized modules that collaborate to accomplish the restoration task. As illustrated in Fig. 2, the system takes three key inputs: a short segment (approximately 3 to 20 s) of the original vocal recording, referred to as **Original Singing A** (in .wav format), which contains the target singer’s performance; the user-specified repair interval defined by **Start Time (ST)** and **End Time (ED)**; and a corresponding musical score, **Score S** (in .musicxml format), aligned in length with the audio segment (typically covering 3 to 8 measures), which contains the desired pitch, phoneme duration, and lyrical content to be restored. The system then outputs a modified version of the audio—**Restored Singing A’** (.wav format)—whose restored portion conforms to the musical intent specified in Score S.

The system integrates three specialized model components:

(1) SOFA (Singing-Oriented Forced Alignment): SOFA is a phoneme-level alignment tool specifically optimized for Mandarin singing voice. In comparison to the Montreal Forced Aligner (MFA) originally adopted by VoiceCraft, SOFA significantly improves alignment accuracy in Chinese singing scenarios. Within this system, SOFA is responsible for aligning **Original Singing A** with the target lyrics **T** extracted from **Score S**, identifying the precise timing of each word or syllable in the audio.

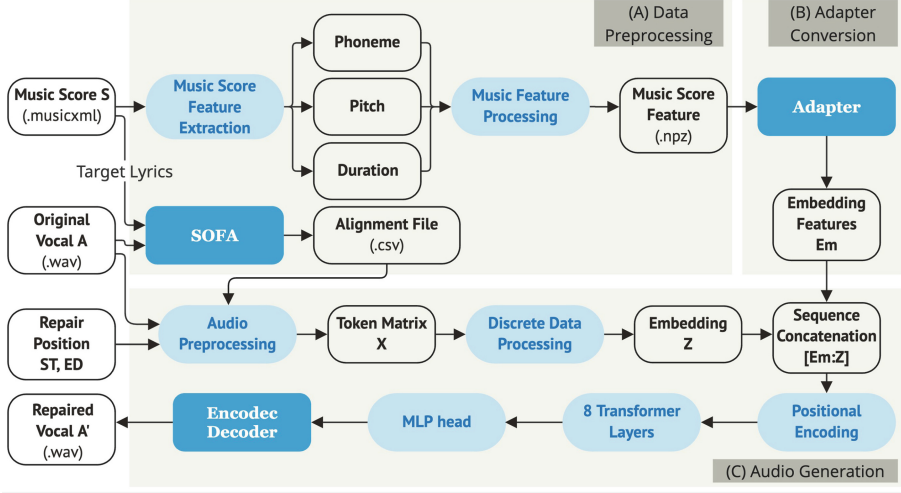


Fig. 2. System Architecture Overview.

(2) **DiffSinger:** DiffSinger is a diffusion-probability-model-based singing voice synthesis system that supports Mandarin. It represents musical data at the phoneme level, enabling detailed modeling of vocal parameters. In this system, DiffSinger is used not to synthesize audio directly, but to extract structured control features—including phonemes, pitch, and duration—from **Score S**. These features serve as inputs for the subsequent Adapter module.

(3) **VoiceCraft:** VoiceCraft is the core generative module of the system. Originally designed as an English zero-shot text-to-speech (TTS) model, VoiceCraft is capable of preserving the original acoustic environment and speaker timbre with high fidelity. In the proposed system, VoiceCraft receives the music-conditioned features—transformed by the Adapter—and generates the desired restoration segment using **Original Singing A** as a reference for speaker style and acoustic context.

The complete processing workflow is divided into three major stages:

- **Data Preprocessing** (Sect. 3.2): This stage includes extracting musical features from **Score S** using the preprocessing module of DiffSinger and performing precise audio-to-lyric alignment between **Original Singing A** and **Target Lyrics T** using SOFA.
- **Adapter Transformation** (Sect. 3.3): The extracted musical features are transformed into embedding tensors compatible with VoiceCraft’s input space.
- **Audio Generation** (Sect. 3.4): VoiceCraft generates the restored segment **Restored Singing A'** by conditioning on the transformed embeddings **Em** and referencing the speaker characteristics in **Original Singing A**.

This modular architecture decomposes the complex audio restoration task into distinct functional units, enabling targeted tuning of individual components and facilitating future system maintenance and scalability.

3.2 Data Preprocessing

Data preprocessing serves as the foundational stage of the system, responsible for transforming raw input into standardized formats suitable for subsequent modules. This stage comprises two main processes: musical score feature extraction and audio-to-lyric alignment, ensuring that the system obtains both complete and accurate conditioning features.

Musical Score Feature Processing. The feature extraction process begins with parsing the input score **Score S**, provided in `.musicxml` format, to retrieve essential musical attributes, including lyrics, pitch, and note duration. The score is processed at the unit of individual lyrics (words or syllables), which are in Mandarin. Pitch is represented using chord notation (e.g., $C\#3/Db3$), with enharmonic equivalence preserved. Note duration is computed in seconds, based on the tempo markings in the score.

To handle special cases in the musical data, we designed a set of rules. For rests, consecutive rest symbols are merged into a single unit with the pitch labeled as “rest.” Corresponding lyrics are marked as “AP” for short pauses (duration $<0.5s$) and “SP” for long pauses ($\geq 0.5s$). For melisma or lyric slurs, when a single lyric unit spans multiple notes: if all notes share the same pitch, their durations are aggregated, and only one pitch value is retained; if pitches differ, all pitch values and their corresponding durations are recorded under the same lyric unit.

The extracted score information is then passed into the Adapter feature composition phase, where lyrics, pitch, and duration values are transformed into formats suitable for system processing. The system first uses DiffSinger’s preprocessing functions to perform an initial transformation: Mandarin lyrics are converted into Chinese phonemes, pitch values are converted to MIDI numbers (ranging from 0 to 127), and pitch-sliding features are added to enhance expressiveness.

To improve model training stability, both pitch and duration features undergo normalization. Pitch features are normalized using Min-Max scaling to the $[0, 1]$ range. Duration features are first log-transformed (to maintain perceptual consistency in musical timing) and then normalized to the same $[0, 1]$ range. All processed features are organized with English phonemes as the fundamental unit and stored in `.npz` format. The normalization reference values are retained to enable denormalization during the generation phase. The final output of this stage is a structured `.npz` file containing musical features.

Audio-to-Lyric Alignment. The audio alignment process aims to establish a precise temporal mapping between the **Original Singing A** and the target

lyrics. We employ the SOFA (Singing-Oriented Forced Aligner) tool for this task. The alignment process consists of three stages:

First is the pre-alignment stage, in which the target lyrics **T** are extracted from the `.npz` score feature file. While there may be content mismatches between the **Original Singing A** and the lyrics in **Score S**—especially in regions requiring restoration—SOFA remains robust in aligning matching segments, even if they appear mid-phrase.

SOFA then converts the extracted Mandarin lyrics into pinyin and generates a plain text label file (`.lab`), which serves as the standardized input for the alignment tool.

Next, in the SOFA alignment stage, the tool applies forced alignment techniques tailored for singing voice to mark the start and end times of each phoneme within the audio. Compared to traditional tools like MFA, SOFA offers greater accuracy for the expressive and elongated nature of singing, particularly in Mandarin.

Finally, in the post-alignment stage, the resulting alignment file (`.TextGrid`) from SOFA is converted into a `.csv` format compatible with the VoiceCraft system. During this conversion, the same Mandarin-to-English phoneme mapping and averaging strategy used in the feature extraction process is applied to ensure consistency in phoneme representation.

Through this data preprocessing pipeline, we establish a complete transformation from musical score to model-ready input, laying the groundwork for the subsequent Adapter conversion and audio generation stages.

3.3 Adapter Transformation

The Adapter module constitutes the core of the proposed system, responsible for transforming representations in the musical feature domain into the text embedding domain used by VoiceCraft. It achieves cross-modal feature mapping, enabling VoiceCraft—originally designed to process only textual inputs—to interpret and generate audio conditioned on musical features. The Adapter is designed to effectively fuse discrete phoneme information with continuous musical features and map them into a unified latent representation compatible with the VoiceCraft model.

Training Dataset. To train the Adapter module, we constructed a dedicated music dataset using real-world song scores. All scores are in monophonic `.musicxml` format, a widely adopted standard for digital sheet music. The use of monophonic lines and single-note representations ensures focus on vocal melody, enabling the model to accurately learn the correspondence between pitch, rhythm, and lyrics.

In total, we collected and processed 302 score segments. The dataset is split randomly into a training set (272 samples) and a validation set (30 samples) with a 9:1 ratio. To assess generalization, we additionally prepared a separate

test set containing 30 score segments from unseen songs, ensuring fair evaluation of the model’s ability to generalize to new musical content.

The dataset includes a diverse range of musical styles to ensure that the trained Adapter module can handle the variety of vocal performances encountered in live concert recordings.

Model Architecture Design. The Adapter employs a deep neural architecture consisting of six functional components: a phoneme embedding layer, a feature processing layer, a feature fusion layer, a positional encoding layer, a main transformation block, and an output projection layer. The hyperparameters are set as follows: hidden dimension = 1024, output dimension = 2048 (to match VoiceCraft’s embedding space), with four residual blocks and dropout probability of 0.1.

Training Strategy and Optimization. The Adapter is trained using mean squared error (MSE) loss, appropriate for regression tasks over continuous embedding spaces. Sequence lengths are constrained between 5 and 500, and dynamic padding with attention masks is applied for efficient batch processing.

We use the AdamW optimizer, which combines adaptive learning rate with weight decay regularization. The learning rate is set to 1e-4, weight decay to 1e-5, and batch size to 16. The model is trained for 50 epochs with early stopping (patience = 5) to prevent overfitting. Learning rate scheduling is handled via ReduceLROnPlateau, which reduces the learning rate when validation loss plateaus. Gradient clipping with a maximum norm of 1.0 is applied to avoid gradient explosion.

Overall, the Adapter successfully enables cross-modal feature transformation from structured musical features to VoiceCraft’s speech embedding space, forming a viable solution for music-conditioned voice generation.

3.4 Audio Generation

The audio generation module serves as the final execution component of the system, responsible for synthesizing the restored singing voice based on musical features transformed by the Adapter. It adopts an autoregressive generation framework based on Transformer architecture, integrating attention mechanisms and structured data processing pipelines to perform voice synthesis and restoration.

Audio Preprocessing Pipeline. The generation process begins with preprocessing. The system compares the user-specified editing positions—**Start (ST)** and **End (ED)**—with the target lyrics extracted from the score **S**. The ST and ED markers may correspond to single words or phrases. A string-matching procedure is employed to locate the corresponding time interval within the lyrics. The system then determines the exact start and end time points of the audio segment that requires editing.

To ensure seamless transitions between edited and unedited regions, the system expands the editing region by 0.1 s on both sides. This temporal extension is constrained to remain within the valid boundaries of the original audio \mathbf{A} .

Subsequently, the original singing audio \mathbf{A} is processed using an Encodec quantizer. The Encodec encoder converts the continuous waveform into a discrete matrix of shape $\mathbf{T} \times \mathbf{K}$, where \mathbf{T} denotes the number of time frames, and \mathbf{K} represents the number of codebooks in residual vector quantization (RVQ). The quantized output is represented as a sequence (X_0, X_1, \dots, X_t) , where each X_i is a \mathbf{K} -dimensional vector containing the quantized token IDs for that frame across the codebooks.

Transformer-Based Generation Architecture. The Transformer generation module takes two types of input: (1) the audio embeddings and (2) the Adapter-derived feature embeddings. For each time step, the embeddings from different codebooks are summed and positional encodings are added using sinusoidal functions to provide temporal context.

The Adapter embeddings include phoneme, pitch, duration, and pitch glide features, already enriched with positional encodings. These two input sequences (audio and control features) are concatenated and jointly fed into the Transformer.

A causal attention mask is employed to preserve autoregressive behavior: the audio portion can attend to all textual features and only to preceding audio tokens, while the textual (Adapter) features have full self-attention. This asymmetric attention scheme allows the Transformer to conditionally generate audio tokens while attending to the entire control feature sequence.

The Transformer consists of 8 layers, each with 16 attention heads. In the output stage, \mathbf{K} independent MLP heads are used—one for each codebook. Each MLP head maps the Transformer output to a logits vector over the token vocabulary of its respective codebook. The logits are then passed through a softmax layer, and token sampling is performed to predict the next token for each codebook.

Audio Decoding and Reconstruction. In the final step, the predicted discrete token IDs are passed to the Encodec decoder. The decoder performs table look-up operations to convert tokens into continuous embedding vectors, which are then processed by a neural vocoder to reconstruct the final waveform. The output is the restored singing voice, denoted as \mathbf{A}' .

4 Experiments

Experimental Setup. The experimental materials in this study consist of musical scores and corresponding clean target vocal recordings that align with the same score range. During the experiment, specific regions within the clean target audio were designated as restoration targets to evaluate the system’s performance.

Since the proposed system employs a masked generation approach, where designated regions are removed and regenerated, the output remains consistent as long as the same restoration region is specified, regardless of whether the input segment contains any actual distortion.

A total of 34 test segments were used in the performance study. Evaluation focused on two key dimensions: pitch accuracy and timing alignment, as these are critical aspects in singing voice restoration.

Three metrics were used to quantify system performance:

1. F0 RMSE (Root Mean Square Error): Measures the deviation of the generated fundamental frequency from the target, indicating pitch accuracy.
2. F0 MAE (Mean Absolute Error): Provides an intuitive estimate of pitch deviation using the average absolute error.
3. Lyric Synchronization Error: Measures the temporal deviation between the generated audio and the phoneme boundaries aligned with the musical score, indicating rhythmic accuracy.

Evaluation Results. Objective evaluation results across the 34 test segments are summarized as follows (Table 1):

Table 1. Summary of Evaluation Results.

Metric	Mean	Std. Dev.
F0 RMSE (<i>Hz</i>)	146.30	137.87
F0 MAE (<i>Hz</i>)	103.64	83.45
Lyric Sync Error (ms.)	687.41	574.09

For pitch accuracy, the F0 RMSE reached 146.30 Hz, and F0 MAE was 103.64 Hz. These values are significantly higher than the typical 20–50 Hz range found in state-of-the-art singing voice synthesis systems. For example, given a reference pitch of C4 (261.63 Hz), the 146.30 Hz error corresponds to nearly half an octave, which explains the perceived pitch inaccuracy. The high standard deviations (137.87 Hz and 83.45 Hz) also indicate substantial variability across test samples—some segments may fall within acceptable bounds, while others exhibit pronounced deviations.

For timing accuracy, the average lyric synchronization error was 687.41 ms, or approximately 0.69 s, which is considerably higher than the 50–200 ms range expected in high-quality systems. The standard deviation of 574.09 ms further reflects large variability in alignment performance, possibly due to differences in phoneme density, note complexity, or background interference.

5 Conclusion

The generative AI-based singing voice restoration system developed in this study offers a novel technological approach for the post-production of concert recordings. However, both subjective and objective evaluations reveal that significant technical challenges remain. Specifically, objective metrics indicate insufficient pitch control accuracy (with an F0 RMSE of 146.30 Hz, equivalent to a deviation of approximately half an octave), limited temporal alignment capability (with an average synchronization error of 687.41 ms), and suboptimal preservation of musical features by the adapter module.

A detailed analysis suggests that these issues may stem from a dimensionality bottleneck in the current adapter architecture. By forcing complex, multidimensional musical features (such as pitch, duration, phonemes, and pitch slides) into a fixed-dimensional text embedding space, critical musical information may be lost during the cross-modal transformation process. This loss of information likely contributes to the degradation of generation quality.

Each proposed improvement pathway introduces distinct technical challenges and varying levels of implementation complexity. As generative AI technologies continue to advance, we anticipate the development of higher-quality, more robust, and more practical solutions for singing voice restoration. Such advancements have the potential to significantly enhance the efficiency and quality of post-production workflows in the music industry, ultimately improving the overall fidelity of concert recordings.

Acknowledgments. This study was funded by the National Science and Technology Council, Taiwan, R.O.C. (grant number NSTC-113-2221-E-030-013 and NSTC-114-2221-E-030-007).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Blaauw, M., Bonada, J.: A neural parametric singing synthesizer modeling timbre and expression from natural songs. *Appl. Sci.* **7**(12) (2017), <https://www.mdpi.com/2076-3417/7/12/1313>
2. Casanova, E., Weber, J., Shulby, C.D., Junior, A.C., Gölge, E., Ponti, M.A.: YourTTS: towards zero-shot multi-speaker TTS and zero-shot voice conversion for everyone. In: Chaudhuri, K., Jegelka, S., Song, L., Szepesvari, C., Niu, G., Sabato, S. (eds.) *Proceedings of the 39th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 162, pp. 2709–2720. PMLR (2022). <https://proceedings.mlr.press/v162/casanova22a.html>
3. Choi, S., Han, S., Kim, D., Ha, S.: Attentron: few-shot text-to-speech utilizing attention-based variable-length embedding. In: *Interspeech 2020*, pp. 2007–2011 (2020). <https://doi.org/10.21437/Interspeech.2020-2096>
4. Elias, I., et al.: Parallel tacotron 2: a non-autoregressive neural tts model with differentiable duration modeling. In: *Interspeech 2021*, pp. 141–145 (2021). <https://doi.org/10.21437/Interspeech.2021-1461>

5. Gong, C., et al.: ZMM-TTS: Zero-shot multilingual and multispeaker speech synthesis conditioned on self-supervised discrete speech representations. *IEEE/ACM Trans. Audio Speech Lang. Process.* **32**, 4036–4051 (2024). <https://doi.org/10.1109/TASLP.2024.3451951>
6. Gu, Y., et al.: Bytesing: a Chinese singing voice synthesis system using duration allocated encoder-decoder acoustic models and wavernn vocoders. In: 2021 12th International Symposium on Chinese Spoken Language Processing (ISCSLP), pp. 1–5 (2021). <https://doi.org/10.1109/ISCSLP49672.2021.9362104>
7. He, J., et al.: RMSSinger: realistic-music-score based singing voice synthesis. In: Rogers, A., Boyd-Graber, J., Okazaki, N. (eds.) *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 236–248. Association for Computational Linguistics, Toronto (2023). <https://doi.org/10.18653/v1/2023.findings-acl.16>, <https://aclanthology.org/2023.findings-acl.16/>
8. Huang, Q., Jansen, A., Lee, J., Ganti, R., Li, J.Y., Ellis, D.P.W.: Mulan: a joint embedding of music audio and natural language. In: *Proceedings of the 23rd International Society for Music Information Retrieval Conference*, pp. 559–566. ISMIR (2022). <https://doi.org/10.5281/zenodo.7316724>
9. Huang, S.F., Lin, C.J., Liu, D.R., Chen, Y.C., Lee, H.Y.: Meta-TTS: meta-learning for few-shot speaker adaptive text-to-speech. *IEEE/ACM Trans. Audio Speech Lang. Process.* **30**, 1558–1571 (2022). <https://doi.org/10.1109/TASLP.2022.3167258>
10. Jeong, M., Kim, M., Choi, B.J., Yoon, J., Jang, W., Kim, N.S.: Transfer learning for low-resource, multi-lingual, and zero-shot multi-speaker text-to-speech. *IEEE/ACM Trans. Audio Speech Lang. Process.* **32**, 1519–1530 (2024). <https://doi.org/10.1109/TASLP.2024.3364085>
11. Li, S., et al.: A survey on cross-modal interaction between music and multi-modal data. *arXiv e-prints arXiv:2504.12796* (2025). <https://doi.org/10.48550/arXiv.2504.12796>
12. Li, X., Wang, Z.: A hmm-based mandarin Chinese singing voice synthesis system. *IEEE/CAA J. Autom. Sinica* **3**(2), 192–202 (2016). <https://doi.org/10.1109/JAS.2016.7451107>
13. Lin, Y.B., Sung, Y.L., Lei, J., Bansal, M., Bertasius, G.: Vision transformers are parameter-efficient audio-visual learners. In: *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2299–2309 (2023). <https://doi.org/10.1109/CVPR52729.2023.00228>
14. Liu, J., Li, C., Ren, Y., Chen, F., Zhao, Z.: DiffSinger: singing voice synthesis via shallow diffusion mechanism. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, no. 10, pp. 11020–11028 (2022). <https://doi.org/10.1609/aaai.v36i10.21350>, <https://ojs.aaai.org/index.php/AAAI/article/view/21350>
15. Lu, P., Wu, J., Luan, J., Tan, X., Zhou, L.: XiaoiceSing: a high-quality and integrated singing voice synthesis system. In: *Interspeech 2020*, pp. 1306–1310 (2020). <https://doi.org/10.21437/Interspeech.2020-1410>
16. Peng, P., Huang, P.Y., Li, S.W., Mohamed, A., Harwath, D.: VoiceCraft: zero-shot speech editing and text-to-speech in the wild. In: Ku, L.W., Martins, A., Srikumar, V. (eds.) *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12442–12462. Association for Computational Linguistics, Bangkok (2024). <https://doi.org/10.18653/v1/2024.acl-long.673>, <https://aclanthology.org/2024.acl-long.673/>
17. Ren, Y., Hu, C., Tan, X., Qin, T., Zhao, S., Zhao, Z., Liu, T.Y.: FastSpeech 2: fast and high-quality end-to-end text to speech. In: *International Conference on Learning Representations* (2021). <https://openreview.net/forum?id=piLPYqxtWuA>

18. Ren, Y., et al.: FastSpeech: Fast, Robust and Controllable Text to Speech. Curran Associates Inc., Red Hook (2019)
19. Shen, K., et al.: Naturalspeech 2: latent diffusion models are natural and zero-shot speech and singing synthesizers. In: International Conference on Learning Representations (2024). <https://openreview.net/forum?id=Rc7dAwVL3v>
20. Sun, Z., Sarma, P., Sethares, W., Liang, Y.: Learning relationships between text, audio, and video via deep canonical correlation for multimodal language analysis. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, no. 05, pp. 8992–8999 (2020). <https://doi.org/10.1609/aaai.v34i05.6431>, <https://ojs.aaai.org/index.php/AAAI/article/view/6431>
21. Tan, X., Qin, T., Soong, F., Liu, T.Y.: A survey on neural speech synthesis. arXiv e-prints [arXiv:2106.15561](https://arxiv.org/abs/2106.15561) (2021). <https://doi.org/10.48550/arXiv.2106.15561>
22. van den Oord, A., et al.: WaveNet: a generative model for raw audio. In: 9th ISCA Workshop on Speech Synthesis Workshop (SSW 9), p. 125 (2016)
23. Wang, J.Y., Lee, H.Y., Jang, J.S.R., Su, L.: Zero-shot singing voice synthesis from musical score. In: 2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 1–8 (2023). <https://doi.org/10.1109/ASRU57964.2023.10389711>
24. Wang, W., Song, Y., Jha, S.: USAT: a universal speaker-adaptive text-to-speech approach. *IEEE/ACM Trans. Audio, Speech and Lang. Process.* **32**, 2590–2604 (2024). <https://doi.org/10.1109/TASLP.2024.3393714>
25. Wang, Y., et al.: Tacotron: towards end-to-end speech synthesis. In: Interspeech 2017, pp. 4006–4010 (2017). <https://doi.org/10.21437/Interspeech.2017-1452>
26. Yu, Y., Tang, S., Raposo, F., Chen, L.: Deep cross-modal correlation learning for audio and lyrics in music retrieval. *ACM Trans. Multimedia Comput. Commun. Appl.* **15**(1) (2019). <https://doi.org/10.1145/3281746>



Revealing the Hidden Temporal Structure of HubertSoft Embeddings Based on the Russian Phonetic Corpus

Anastasia Ananeva¹(✉) , Anton Tomilov² , and Marina Volkova²

¹ ITMO University, Saint Petersburg, Russia
464995@niuitmo.ru

² STC-innovations Ltd., Saint Petersburg, Russia
{tomilov,volkova}@speechpro.com

Abstract. Self-supervised learning (SSL) models such as Wav2Vec 2.0 and HuBERT have shown remarkable success in extracting phonetic information from raw audio without labelled data. While prior work has demonstrated that SSL embeddings encode phonetic features at the frame level, it remains unclear whether these models preserve temporal structure, specifically, whether embeddings at phoneme boundaries reflect the identity and order of adjacent phonemes. This study investigates the extent to which boundary-sensitive embeddings from HubertSoft, a soft-clustering variant of HuBERT, encode phoneme transitions. Using the CORPRES Russian speech corpus, we labelled 20 ms embedding windows with triplets of phonemes corresponding to their start, centre, and end segments. A neural network was trained to predict these positions separately, and multiple evaluation metrics, such as ordered, unordered accuracy and a flexible centre accuracy, were used to assess temporal sensitivity. Results show that the model achieves up to 53% ordered accuracy, indicating strong temporal awareness, while unordered accuracy exceeds 90%, confirming robust phonetic content encoding. Confusion patterns further suggest that the model encodes articulatory detail and coarticulatory effects. These findings contribute to our understanding of the internal structure of SSL speech representations and their potential for phonological analysis and fine-grained transcription tasks.

Keywords: Temporal embedding structure · phonetic embedding analysis · self-supervised learning · HubertSoft

1 Introduction

Self-supervised learning (SSL) models such as Wav2Vec 2.0 [1] and HuBERT [2] have significantly advanced the field of speech representation learning by extracting rich phonetic information without labelled data. The embeddings received usually contain a large amount of information which is used in deepfake detection [4]. These models have demonstrated impressive performance on tasks

including phoneme classification [3], automatic speech recognition [5] and articulatory feature prediction [6]. However, it remains underexplored how these embeddings encode phoneme boundaries and whether the temporal structure is also encoded.

It has been shown that SSL embeddings encode temporally localized phonetic cues accessible through supervised probing [7]. For instance, linear or shallow MLP probes have effectively predicted phonetic features such as place, manner, and voicing at the frame level [8, 9]. These works use Wav2Vec 2.0 to confirm that transitions in the embedding space align closely with phoneme boundaries, offering indirect evidence of temporal structure encoding. Other approaches track articulatory feature trajectories or analyse embedding differences at phoneme transitions, further supporting the presence of boundary-sensitive representations [10].

Phonetic features are also reflected in the embeddings received with HuBERT [3, 13]. HuBERT (Hidden-Unit BERT) is a self-supervised speech representation model that relies on a masked prediction task, similar to masked language modeling in NLP. The model is trained to predict the identity of masked audio frames using discrete labels obtained from an offline clustering step (k-means) applied to acoustic features. During training, HuBERT learns contextualized representations by encoding the speech signal with a convolutional feature extractor followed by a Transformer encoder. These representations have been shown to capture various levels of phonetic, prosodic, and even semantic information, depending on the layer and training stage.

However, these studies focus mainly on some phonetic features encoded in the embeddings. For example, HuBERT embeddings differentiate vowel quality by encoding distinctions in vowel height, backness, and roundedness. They also effectively represent consonantal articulation features such as place and manner, voicing, nasality, and the stop–fricative contrast. Beyond segmental properties, HuBERT embeddings have been found to reflect prosodic features like pitch, stress, and intonation, and are sensitive to phonetic context effects, including coarticulation.

In addition, there is growing evidence that both self-supervised and supervised models can be effectively used for phoneme boundary detection. Self-supervised models have been shown to capture abrupt spectral or articulatory transitions in their embeddings, which align closely with phoneme boundaries [11]. Techniques such as comparing adjacent frame representations or applying peak detection over embedding dissimilarities allow for boundary identification without explicit labels, suggesting that SSL models implicitly encode segmental structure alongside phonetic content. Complementing this, supervised approaches such as those using Connectionist Temporal Classification (CTC) loss have framed the task as aligning sequences of phoneme pairs to speech. This enables models to focus on transition regions and predict boundaries with high temporal precision [12]. Together, these lines of work highlight the capacity of learned representations to reflect temporal dynamics in speech.

Despite this progress, to the best of our knowledge, no prior study has directly tested whether SSL embeddings encode the identity of the phoneme that begins or ends a segment encoded. The existing literature focuses predominantly on general framework phonetic classification or the analysis of internal embedding structure.

This gap is especially pertinent given the growing interest in understanding how SSL models internalize sub-segmental linguistic structure. Notably, there is also a lack of probing studies involving HuBERTSoft [14], which is a modification of HuBERT that replaces hard clustering of latent speech representations with soft posterior distributions over learned acoustic units. Unlike the original HuBERT, which uses k-means to assign each frame to a single cluster, HuBERTSoft outputs a probability distribution over clusters, allowing for richer, more nuanced modeling of acoustic patterns.

The current work aims to fill this gap by evaluating whether boundary-frame embeddings from HubertSoft can be used to predict the phoneme that begins or ends a segment. By focusing on supervised probing of initial and final frames at phoneme boundaries, we offer a novel perspective on the structure encoded within self-supervised speech embeddings and shed light on the representations' utility for fine-grained phonological tasks.

2 Material

The experiments were conducted using the CORPRES [15] Russian speech corpus. The dataset contains recordings of 4 male and 4 female speaker annotated on multiple linguistic levels. The total duration is 30h. Phonetic labelling was performed by trained phoneticians, ensuring a high degree of reliability. The corpus uses a phonemic annotation scheme largely aligned with the IPA system for Russian [16], with a few systematic modifications. Vowels are annotated with numeric indices that indicate their position relative to the stressed syllable: for example, a0 denotes a stressed /a/, a1 and a2 indicate vowels in pre-stressed positions (closer and further from the stressed syllable, respectively), and a4 marks a post-stressed vowel. For other vowels numbers 0, 1 and for are used. For the vowel /i/ the symbol y is used. This labeling provides a finer-grained representation of Russian vowel reduction patterns, which are known to be stress-dependent. The palatalization symbol ^j is replaced with '. Additionally, certain consonants are represented using simplified or alternative symbols: for instance, the affricate ts is denoted as c, and similar substitutions (t^j - ch, f- sh, f^j - sch, ʒ- zh) are applied consistently throughout the corpus to maintain compatibility with the transcription system used during preprocessing.

However, in certain instances, particularly at the boundaries between acoustically similar sounds, such as between two vowels or a vowel and a consonant, the segmentation may be imprecise. In such cases, the boundary is typically placed approximately at the midpoint of the acoustic transition between the sounds [17]. The hand-labelled recordings from eight speakers were utilized for embedding extraction with the HubertSoft model. For embedding extraction,

the HubertSoft model was used with frozen weights, while only the classification heads were trained. Initially, embeddings were extracted from the recordings and then averaged within phonetic segment boundaries, providing a compact representation of each labelled sound. These averaged embeddings were used in a subset of experiments to establish baseline performance. The symbols for various sounds were also taken from the CORPRES.

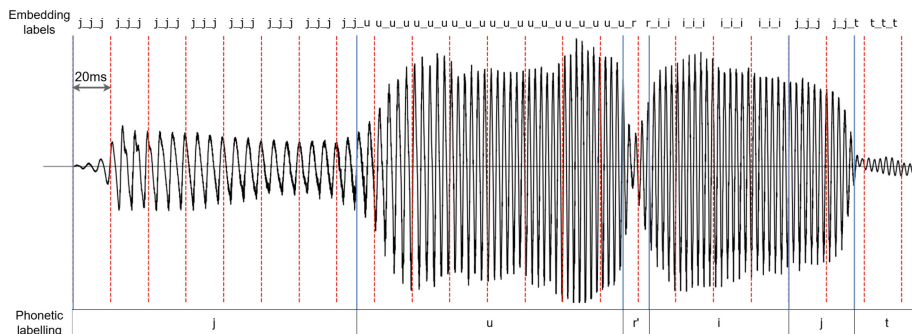


Fig. 1. Example of speech signal segmentation into 20ms frames with corresponding phonetic labels and frame-wise assignments.

To move towards more fine-grained, automatic analysis suitable for speech transcription tasks, we also extracted non-averaged embeddings. In this setup, each embedding represented a 20 ms window of the audio signal, as shown in Fig. 1, without overlap. This approach preserved high temporal resolution, which was essential due to the presence of very short phonetic events (as brief as 5 ms) in the corpus. Such short phoneme realizations are usually unstressed vowels or /r/ and they appear due to the fluency of speech. They may not be pronounced properly, but still are labelled to reflect the phonetic composition of speech.

Each embedding was assigned a triplet label indicating the phoneme present at its beginning, centre, and end, derived from the time-aligned phonetic annotations. For the start and end positions, the label was determined by the phoneme occupying the first or last millisecond of the 20ms window, respectively, so no minimum duration threshold was applied. The central label was assigned based on the phoneme that occupied the majority of the middle portion of the window.

Table 1 gives some examples of possible labels. This labelling scheme allowed us to capture phonetic transitions within the window. In most cases, all three sub-labels were identical, suggesting that the window was fully inside a single phoneme (typically its central part). However, a substantial number of embeddings spanned phoneme boundaries, where the centre label often matched either the start or end label. These boundary-spanning embeddings are particularly informative for analysing whether the learned representations reflect the temporal structure of the speech signal.

Table 1. Examples of labels.

Central embedding	Border embedding	Two-border embedding
a_a_a, p_p_p	a_p_p, p_p_a, a_a_p	a_p_a, a_p_s

This research focuses mainly on the border embeddings, as they represent a part of a non-homogeneous signal, in which the sequence of sounds matters. Two-border embeddings were excluded as there is a small number of them.

3 Method

Our previous experiments demonstrated that HuBERTSoft embeddings encode substantial phonetically relevant information. Specifically, we evaluated their capacity to represent phonetic distinctions through a consonant classification task using averaged embeddings (i.e., embeddings averaged over the duration of each phoneme). A 4-layer feedforward neural network was successfully trained to classify these averaged embeddings into consonant categories. The model architecture consisted of four fully connected layers with 512, 256, 128, and 64 neurons respectively, each followed by ReLU activation and dropout ($p = 0.1$). The recordings of 4 speakers from the corpus were used to train the model. Evaluation was conducted using the other for speakers. Based on these findings, we adapted the same architecture for the classification of non-averaged, boundary-sensitive embeddings, with separate output heads corresponding to the start, centre, and end positions of each 20 ms window.

To assess generalization, a speaker-independent evaluation setup was used: recordings from four speakers in the corpus were allocated for training, and the remaining four speakers were used for testing.

For training, only embeddings whose associated labels contained phonemes from a predefined target list were selected, which allowed us to limit the target group and not to use the whole phoneme inventory of the language. The predefined target list included the following phonemes, grouped as follows: “a0”, “a1”, “a2”, “a4”, “i0”, “i1”, “i4”, “o0”, “y0”, “y1”, “y2”, “u0”, “u1”, “u2”, “e0”, “l”, “l”, “m”, “m”, “n”, “n”, “p”, “p”, “t”, “t”, “k”, “k”, “s”, “s”, “f”, “f”, “h”, “h”, “b”, “b”, “d”, “d”, “g”, “g”, “c”, “ch”, “sh”, “v”, “v”. To address class imbalance, some vowels (e.g., unstressed e and o) and rare consonants (e.g., zh’) were excluded. The maximum number of instances of one phoneme type per speaker was limited to 1000.

After this filtering, both the training and test sets contained embeddings from 4 speakers, with up to 1000 realizations per phoneme type per speaker. The total number of phoneme types varied between groups according to the above criteria. Each selected embedding was paired with a triplet label marking the phonetic class present at the beginning, centre, and end of the 20 ms window.

The network was trained using the Cross Entropy loss function and the AdamW optimizer. The loss function was computed as the sum of cross-entropy

losses across the three output positions. Some hyper-parameters including learning rate of 0.001, the dropout size of 0.1 and weight decay of 0.0001 were experimentally adjusted. The model has been training for 10 epochs.

To evaluate whether the HubertSoft embeddings capture the temporal structure of the signal, we trained a neural network to predict phoneme labels corresponding to the beginning, middle, and end parts of a 20 ms segment. Evaluation was conducted using multiple metrics designed to assess different aspects of temporal representation. **Ordered accuracy** was computed to measure the proportion of predictions where all three positions (start, centre, end) were correctly classified in the correct sequence, reflecting the model’s ability to recognize and preserve temporal order in the embedding space.

To complement this, we introduced **unordered accuracy**, where predictions were considered correct if they matched the set of target labels regardless of order. While temporal order is essential in speech, this metric helps isolate and evaluate whether the embeddings encode the phonetic content itself—independently of its alignment in time. In other words, unordered accuracy assesses whether the correct phonemes are present in the representation, even if the temporal structure is not perfectly captured. This provides insight into the extent to which the embedding space encodes phonetic identity versus temporal sequencing, which is valuable for downstream tasks where either or both dimensions may be important.

To better evaluate the model’s ability to localize phonetic boundaries while allowing some ambiguity in the center position, we introduce another metric: **ordered flexible centre accuracy**.

This metric relaxes the strict requirement of full label sequence matching by enforcing correctness at the boundaries while allowing the centre prediction to vary within a constrained range. Specifically, a prediction is considered correct if:

- the predicted start and predicted end labels exactly match the corresponding true start and end labels, and
- the predicted centre label matches either the true start or true end label.

Table 2 summarizes all possible labels considered correct for each metric. Notably, unordered accuracy captures the presence of labels regardless of their order.

4 Results

Classification experiments were conducted to evaluate the model’s ability to recognize phonemic content from embeddings. Several phoneme groups were tested, with the most extensive group comprising 6 stressed vowels and 35 consonants, both palatalized and non-palatalized. This group provides a representative sample of the sound inventory and includes diverse acoustic features. The results for this group are presented in Table 3.

Table 2. Examples of what is considered to be a correct label for different metrics.

Metric	True label	Correct predicted label(s)
Unordered accuracy	a_p_p	a_p_p, a_a_p, p_a_p, p_p_a, a_p_a, p_a_a
Ordered accuracy	a_p_p	a_p_p
Ordered flexible centre accuracy	a_p_p	a_p_p, a_a_p

Table 3. Classification results for the group of vowels and consonants

Metrics	Value
Ordered accuracy	0.5312
Unordered accuracy	0.9069
Ordered flexible centre accuracy	0.7645
Start accuracy	0.8823
Centre accuracy	0.6277
End accuracy	0.8563

To establish a baseline for ordered accuracy, we simulated a model that correctly identifies the set of phonemes in a given embedding with 0.9 probability, but assigns them to positions (start, centre, end) at random. Thus, for the target label “a_p_p” the model will predict the set a, p, p with probability of 0.9, but the order will be random. This simulates a model that has access to phonetic content but lacks temporal resolution. The 0.9 parameter reflects the likelihood of correctly identifying the phonemes involved in the triplet, rather than predicting each position independently. Under this assumption, the expected ordered accuracy is approximately 0.22, illustrating how even accurate phoneme identification is insufficient without reliable temporal assignment. This baseline highlights the added value of temporal structure in the learned embeddings.

In comparison, our model achieves an ordered accuracy of 0.53 across all phoneme groups. This result significantly exceeds the baseline, indicating the model’s capacity to infer not only phoneme identities but also their temporal arrangement within the segment.

The unordered accuracy, which evaluates phoneme presence regardless of position, is approximately 0.91 for this group. The clear gap between unordered and ordered accuracies suggests that the model captures some degree of temporal structure inherent in the embeddings.

To further investigate positional sensitivity, we analysed classification accuracy for each of the three phoneme positions: start, centre, and end. The central position revealed a noticeable drop in value (0.63), suggesting that the model frequently confuses centre phonemes with those in the start or end positions. In contrast, the start and end positions yielded higher accuracies (0.88 and 0.86,

respectively), likely due to more distinct acoustic transitions at the segment boundaries.

The ordered flexible centre accuracy metric, which tolerates minor shifts around the central position, reaches 0.77. This result supports the hypothesis that embeddings preserve temporal phoneme structure to a meaningful extent.

4.1 Performance Across Sound Groups

A consistent pattern emerged while comparing performance across different sound groups, such as vowels, voiced and devoiced plosives, fricatives, palatalized and non-palatalized consonants. Including vowels in the groups for start and end accuracy was important because phoneme boundaries often occur between vowels and consonants in natural speech. By considering vowels alongside consonants at these boundaries, we effectively increase the number of examples where phoneme transitions occur, which helps the model learn and be evaluated on detecting these transitions more reliably.

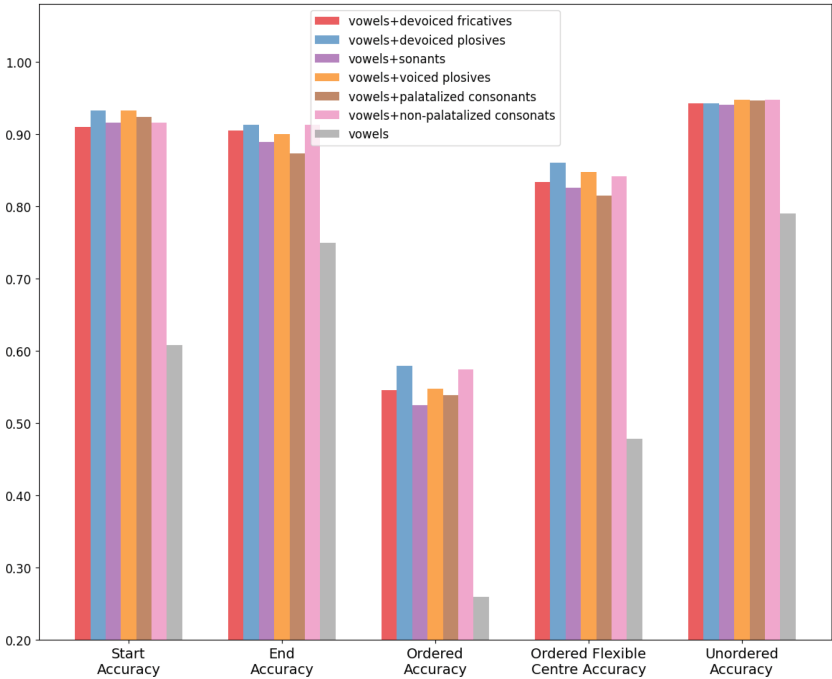


Fig. 2. Comparison of classification accuracies across sound types.

All metrics tended to be highest for the group of voiceless plosives and lowest for sonants and vowels as it is shown in Fig. 2. The group of various vowels, both stressed and not stressed, showed the lowest results. This can be attributed to

the presence of clear transitions between segments with periodic structure (e.g., vowels) and aperiodic segments (e.g., plosives), which creates sharper acoustic boundaries and facilitates detection.

A more detailed examination of end-position predictions, particularly for vowels and devoiced plosives, highlights some systematic patterns. For instance, the palatalized plosive labeled as p' is frequently confused with its hard counterpart p as can be seen in Fig. ¹ 3. In Russian, $/p^j/$ and $/p/$ are considered a phonemic pair, and this confusion is phonetically plausible: both begin with a closure, and their primary distinction lies in the release phase or even the onset of the following vowel. Since the model classifies the final phoneme in the sequence (which is the onset of the next segment) the embedding appears to encode primarily the closure information, making p' and p difficult to distinguish at this point.

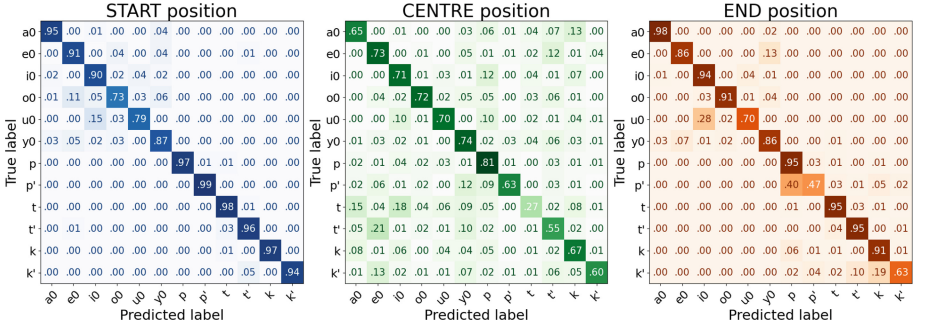


Fig. 3. Confusion matrices for start, center and end position for the group of plosives and vowels.

A similar trend is observed for the $k-k'$ pair. Interestingly, the $t-t'$ contrast does not exhibit the same level of confusion. Although $/t^j/$ is categorized as a plosive in Russian, acoustically it tends toward a fricative-like realization, often approximating $[\text{ts}]$ [18]. This produces a noisier closure phase, which helps the model differentiate it even at the start of the segment. Thus, acoustic differences rooted in allophonic variation seem to impact the model’s ability to distinguish palatalized consonants.

Another example of differentiation between the beginning and the end of can be found in the comparison of the affricate ts to plosive t and fricative s , which have a common place of articulation. The results showed the high similarity of ts sound to the s sound in the start position, which corresponds to the ending of the first sound. In contrast, it becomes closer to the t sound at its beginning, because both have a closure. This shows us that the model classifies the parts of the embedding independently and is definitely capable of distinguishing between ways of articulation.

¹ The symbols for sounds in figures are taken from the CORPRES labelling.

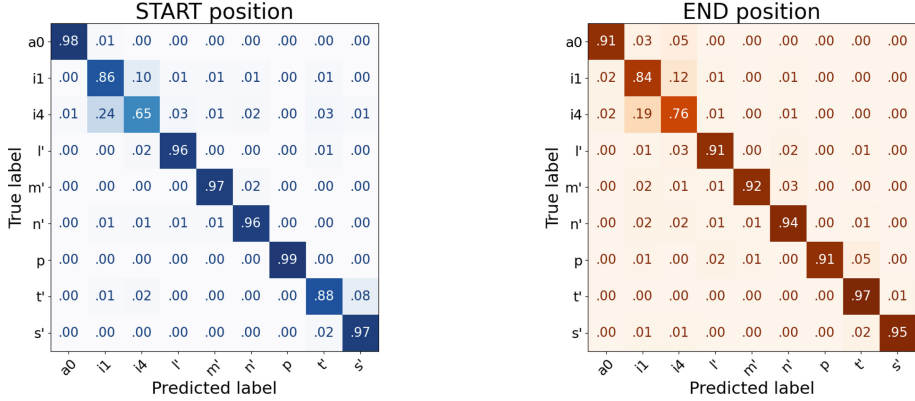


Fig. 4. Confusion matrices for start and end position for the group of palatalized consonants and a0.

A comparable pattern of context-dependent variation is observed in the embeddings for the vowel *a*, particularly in the end position after palatalized consonants. The confusion matrices in Fig. 4 show that while *a* (a0 in this case²) is recognized with high confidence at the start of the segment (0.98), its representation becomes more ambiguous at the end (0.91). Specifically, although the diagonal value for *a0* remains relatively high, there is a noticeable increase in off-diagonal activations toward other vowel classes, *i1*³ and *i0* specifically. This aligns with known coarticulatory effects in Russian, where a preceding palatalized consonant can front the articulation of the following vowel which effectively shifts *a* toward an [i]-like quality [19]. Since the end-position predictions target the onset of the following segment, the model appears to encode this transitional state in the vowel embedding. The fact that this shift is not evident in the start-position matrix further supports the interpretation that the embedding space captures dynamic phonetic interactions rather than static phoneme identity alone. Thus, the model’s behaviour provides indirect evidence of temporal encoding of coarticulatory influence, particularly for vowel segments in soft-consonant contexts.

4.2 Application of the Model

The presence of temporal structure in the learned embeddings enables precise phonetic labelling within short audio segments. Importantly, this enhancement in temporal resolution was achieved without full retraining of the HuBERT model, highlighting the efficiency of the approach. To explore this potential, we applied our model, trained using embeddings from the central and boundary regions of segments, to a short recording of approximately 350 ms in length which contains

² 0 denotes a stressed vowel position.

³ 1 denotes a pre-stressed vowel position.

two borders between sounds. This recording contained three consecutive sounds: *i*l, *l*, and *a*0.

It is important to note that the model was pre-trained exclusively on stressed vowels. Therefore, it cannot reliably distinguish between different stress positions, such as *i*0 and *i*l, which limits its ability to differentiate between pre-stressed and stressed vowel variants.

The model achieved an ordered accuracy of 0.88 when labelling the segment, indicating strong alignment with the expected phoneme sequence. Figure 5 presents the probability distribution across all three phoneme positions over 17 frames. A clear boundary between the first and second sound is visible around frame 3, while frames 8 and 9 capture the transition from *l* to *a*, corresponding to the end of *l* and start of *a*, respectively.

In fact, the only mistake that was achieved is that the border between *i*0 and *l* should have been in the frame number 4 according to the initial labelling. The smoothness of transition between a vowel and a sonant makes it impossible to define the precise location of a boundary.

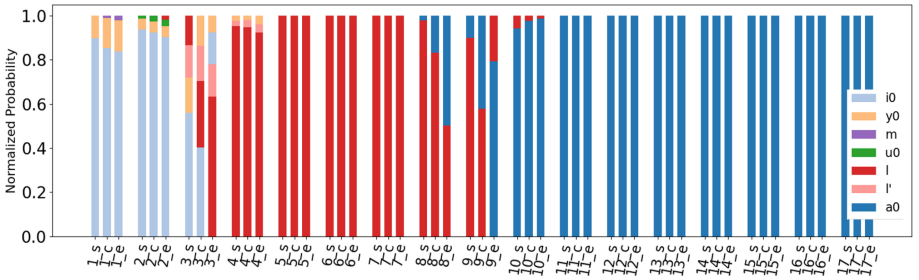


Fig. 5. The distribution of predicted probabilities for target start(s), center(c) and end(e) sounds across 17 frames for each phoneme position in the sound sequence “il l a0”.

This example illustrates the model’s effectiveness in capturing temporal dynamics within embeddings and demonstrates the feasibility of using such representations for fine-grained phonetic segmentation and labelling.

5 Discussion and Conclusion

This study explored the extent to which self-supervised speech embeddings, produced by HubertSoft, encode information related to segment boundaries and temporal structure. The results demonstrate that they not only retain phoneme identity but also encode temporal order with a level of precision sufficient for boundary-aware classification. The model achieved high unordered accuracy, indicating that phonetic content is consistently preserved. Crucially, its ordered and flexible-centre accuracies significantly outperformed chance baselines, suggesting a strong sensitivity to phoneme positioning within the temporal window.

Detailed analysis across phoneme groups revealed that boundary encoding is particularly robust for consonants with distinct acoustic transitions, such as voiceless plosives. In contrast, vowels and sonorants showed lower accuracy, likely due to their smoother spectral characteristics and higher susceptibility to coarticulation. Moreover, some phonetic effects found on the boundaries are also reflected in the embeddings.

These findings show that self-supervised embeddings do not merely encode static phonetic labels but also reflect temporal and contextual dependencies relevant for linguistic analysis. Moreover, the proposed triplet-label probing setup offers a new way to assess the temporal encoding capacity of SSL models.

Summing up, HubertSoft embeddings prove capable of encoding phonemic timing and boundary information, reinforcing their utility for downstream phonological tasks and motivating further work into context-aware, temporally structured speech representations. In future studies, the proposed approach may be applied to explore the internal structure of speech sounds in more detail, potentially contributing to more precise and linguistically informed phonetic labelling. Moreover, future research may consider not only border sequences with two different types of sounds, but also central sequences, as well as segments that contain two boundaries, to better capture the dynamics of phonemic transitions within broader contexts.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Baevski, A., Zhou, Y., Mohamed, A., Auli, M.: wav2vec 2.0: a framework for self-supervised learning of speech representations. *Adv. Neural Inf. Process. Syst.* **33**, 12449–12460 (2020)
2. Hsu, W.-N., Bolte, B., Tsai, Y.-H.H., Lakhota, K., Salakhutdinov, R., Mohamed, A.: HuBERT: self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Trans. Audio Speech Lang. Process.* **29**, 3451–3460 (2021)
3. Martin, K., Gauthier, J., Breiss, C., Levy, R.: Probing self-supervised speech models for phonetic and phonemic information: a case study in aspiration. In: *Proceedings of the Interspeech*, pp. 251–255 (2023). <https://doi.org/10.21437/Interspeech.2023-2359>
4. Stan, A., Combei, D., Oneata, D., Cucu, H.: TADA: training-free attribution and out-of-domain detection of audio deepfakes. *arXiv preprint arXiv:2506.05802* (2025)
5. Schneider, S., Baevski, A., Collobert, R., Auli, M.: wav2vec: unsupervised pre-training for speech recognition. *arXiv preprint arXiv:1904.05862* (2019)
6. Cho, C.J., Watanabe, S., Zhang, Y., Karita, S., Wang, C., Yan, Y.: Evidence of vocal tract articulation in self-supervised learning of speech. In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1–5. IEEE (2023)

7. English, P.C., Kelleher, J., Carson-Berndsen, J.: Domain-informed probing of wav2vec 2.0 embeddings for phonetic features. In: Proceedings of the 19th SIG-MORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology, pp. 83–91 (2022)
8. English, P.C., Kelleher, J.D., Carson-Berndsen, J.: Discovering phonetic feature event patterns in transformer embeddings. In: Proceedings of the Interspeech (2023)
9. Pasad, A., Chou, J.C., Livescu, K.: Layer-wise analysis of a self-supervised speech representation model. In: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 914–921. IEEE, Cartagena (2021)
10. English, P.C., Shams, E.A., Kelleher, J.D., Carson-Berndsen, J.: Following the embedding: identifying transition phenomena in wav2vec 2.0 representations of speech audio. In: ICASSP 2024 – IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6685–6689. IEEE, Seoul (2024)
11. Kreuk, F., Keshet, J., Adi, Y.: Self-supervised contrastive learning for unsupervised phoneme segmentation. arXiv preprint [arXiv:2007.13465](https://arxiv.org/abs/2007.13465) (2020)
12. Arik, S.Ö., et al.: Deep voice: real-time neural text-to-speech. In: International Conference on Machine Learning, pp. 195–204. PMLR (2017)
13. Wells, D., Tang, H., Richmond, K.: Phonetic analysis of self-supervised representations of English speech. In: Proceedings of the Interspeech, pp. 3583–3587. ISCA, Incheon (2022)
14. Van Niekirk, B., Carbonneau, M.A., Zaïdi, J., Baas, M., Seuté, H., Kamper, H.: A comparison of discrete and soft speech units for improved voice conversion. In: ICASSP 2022 – IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 6562–6566. IEEE, Singapore (2022)
15. Skrelin, P., Volskaya, N., Kocharov, D., Evgrafova, K., Glotova, O., Evdokimova, V.: CORPRES: corpus of Russian professionally read speech. In: Sojka, P., Horák, A., Kopeček, I., Pala, K. (eds.) TSD 2010. LNCS (LNAI), vol. 6231, pp. 392–399. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-15760-8_50
16. Yanushevskaya, I., Bunčić, D.: Russian. J. Int. Phon. Assoc. **45**(2), 221–228 (2015)
17. Heselwood, B.: Phonetic Transcription in Theory and Practice. Edinburgh University Press (2013)
18. Bondarko, L.V., Verbitskaya, L.A.: On the markedness of the palatalization feature in Russian consonants. [O markirovannosti priznaka myagkosti russkikh soglasnykh]. STUF–Lang. Typol. Univ. **18**(1–6), 117–124 (1965)
19. Tananayko, S.O.: Acoustic characteristics of vowels after palatalized consonants under interference from related languages (based on Russian speech by Polish speakers). [Akusticheskie kharakteristiki glasnykh posle myagkikh soglasnykh v usloviyakh interferentsii rodstvennykh yazykov (na materiale russkoy rechi polyakov)]. Author’s abstract of Candidate of Philological Sciences Dissertation, St. Petersburg, 15 p (1993)

Natural Language Processing



Analyzing Web-Scraped and Generated Inputs for Automatic and Scalable Intent Classification

Philine Kowol and Stefan Hillmann

Quality and Usability Lab, Technische Universität Berlin, Berlin, Germany
{p.goerzig,stefan.hillmann}@tu-berlin.de

Abstract. Many real-world NLU deployments require coverage of thousands of dynamically evolving topics beyond what manually curated data can sustain. This paper presents an automated pipeline deployed at Technische Universität Berlin (TU Berlin) to extract and train an intent classification model over 2,600 topics with zero manual annotations. We evaluate the effectiveness and generalizability of this pipeline using both curated and LLM-generated data, showing that our automatically trained models surpasses manually engineered systems in some dimensions. While lacking standardized benchmarks, we use a combination of internal evaluation strategies to provide a well-rounded assessment of model robustness. These findings support the use of scalable, self-maintaining NLU systems in complex and dynamic information environments.

Keywords: Intent classification · Dialogue systems · Web scraping · NLU · Automatic data generation

1 Introduction

Intent classification is a foundational component of task-oriented dialogue systems. An intent describes the purpose or goal that a user pursues with an utterance directed to dialogue system. E.g., the input “How can I re-register for the summer semester?” signals the intent *re-registration for studies*. Typically, automatic intent classification relies on manually curated utterances and intents. However, this strategy quickly becomes unmanageable when scaling to hundreds or thousands of potential user needs and topics, especially in domains where content evolves and changes constantly.

We address this challenge in the context of a university chatbot developed to support the student administration office of Technische Universität Berlin (TU Berlin). The information the office employees rely on is structured into a large, dynamically evolving web hierarchy. Initial efforts to develop a traditional NLU system covered only a small portion of the domain. They failed to scale due to a lack of labeled queries and data security restrictions on using real historical queries. This led us to design a hybrid system with a manually curated main-NLU and a backup-NLU trained solely on data extracted from the website’s content and structure.

This paper presents an in-depth analysis of the backup-NLU pipeline. We evaluate how to build the best features from the extracted data, the performance of using an LLM to generate training data, how to test the system performance without an established dataset, and compare its performance to our traditional main-NLU.

Our work contributes to developing scalable and robust NLU systems that function with minimal manual intervention. This is important for making real-world conversational systems more maintainable and adaptive.

2 Related Work

Traditional intent classification pipelines rely on manually labeled data, which limits scalability in dynamic, high-coverage domains such as university websites.

Recent work addresses this through automated or semi-automated training data generation [3]. Transformer-based approaches have become popular, combining structured content with synthetic queries, including question generation pipelines for software [1] and healthcare bots [8].

Besides the generation of additional or exclusively used training data, Rodrigues et al. shows that more granular intents can be of benefit for intent classification. Even if we not generate more granular intents, we guess that generating training material for very granular intents in our use-case is a valid approach.

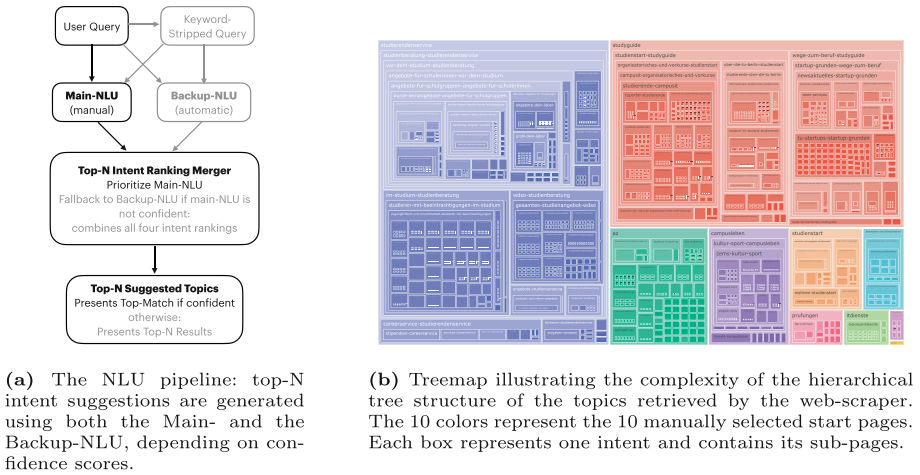
Using small models for intent classification based on embeddings (e.g., generated with BERT-like models) is more efficient and reliable in case of sufficient training examples [2]. Thus, we combine different approaches for automatic training data generation or even just extraction with efficient embedding-based classification approaches [4].

Our work builds on these trends by demonstrating a fully automated, self-maintaining intent classification system based solely on scraped website content, evaluated across thousands of real-world intents.

3 System Overview

The chatbot system builds on previously published work describing an architecture for multilingual intent classification in a university context [6, 7]. As stated in our previous work, the system consists of two NLU components using Rasa NLU with a DIET classifier [4] with LaBSE embeddings [5]. The focus of this paper is not the choice of the model but the choice of training data.

- **Main-NLU:** Covers 62 core topics using manually labeled utterances curated over several years. These include official data from university staff and high-precision examples added based on chatbot logs and iterative testing.
- **Backup-NLU:** Dynamically changing and adapting and trained entirely on automatically extracted content (titles, keywords, sentences) from a large subset of TU Berlin’s website, comprised of over 2,600 topics. This model allows the chatbot to scale beyond the limited manually supervised scope of the main-NLU.



extracted features used as training utterances include sentences, page titles, linking anchor texts, and TF-IDF-based keywords.

4 Data Sources

Our evaluation uses three distinct datasets in multiple variations which are described in the following. Additionally, Table 2 provides an overview on the the contained data and the usage within the described work.

Table 2. Overview of datasets used in training and evaluation.

Dataset	#Intents	#Utterances	Source	Purpose
Manual-62	62	3,005	experts + team	Evaluation
LLM-62	62	~1,000	LLaMA 3	Evaluation
LLM-X	X	~16 * X	LLaMA 3	Evaluation
LLM-Full	2,658	41,858	LLaMA 3	(Full) Evaluation
AZ/Extracted	2,658	varies	Scraped web data	Training
Generated	2,658	varies	LLaMA	Training

4.1 Test Sets

The two test sets are used in different scenarios. The Manual-62 test set is limited to 62 selected intents, while the LLM-Generated test set can scale to the number of intents used in a specific scenario.

Manual-62. The Manual-62 test set includes 3,005 utterances spanning 62 intents. The examples originate from two sources: official query examples provided by TU Berlin’s student administration office, and internally curated utterances based on usage observations and iterative testing. A portion of this data was originally generated via LLMs and later validated and filtered by human assistants. This test set is the training set of the main-NLU and is also used to evaluate the quality of different versions of the backup-NLU.

LLM-Full. We generated 41,858 utterances across 2,676 topics using LLaMA 3 (70B instruct) as LLM. For each part of an webpage that addresses a topic, prompts asked the LLM to list natural-language questions that are directly answerable from the given text. Although the data is synthetically generated, evaluation results indicate that performance trends on this dataset align with those on the Manual-62 set, despite lower overall difficulty.

LLM-62. To compare the LLM-Full and the Manual-62 test sets, we scaled LLM-Full down to the same set intents as covered by Manual-62 and ignore all other generated data from LLM-Full. The result is the LLM-62 test set.

LLM-X. For some scenarios, the number of tested intents needs to be adapted to the number of intents in the respective training set (described below). Here, X represents the number of intents (including the generated example utterance) in that dataset.

4.2 Training Data

AZ: Extracted Training Data. The training data for the backup-NLU was generated by scraping and processing TU Berlin’s website content. The automatic scraping process starts initially on the A-Z page of TU Berlin’s administration office. This page represents an index of relevant topics and we address to it in the following by “AZ”. The backup-NLU is trained on this AZ dataset. Most further experiments use a variation of this dataset. Features extracted for each page included: titles (HTML page titles and link anchors), sentences extracted from website text and keywords gained from the text through TF-IDF¹ computed over the entire corpus of the text from all scraped web-pages.

Pages were filtered using heuristics to limit depth and descendant count. Ultimately, 2,658 pages were included, averaging 16 utterances (i.e., sentences) per topic.

AZ-62. AZ-62 is a variation of the backup-NLU that was only trained on the manually selected 62 “main”-intents from Manual-62 using the automatically extracted data (as previously described for AZ). We are using this model to compare the performance to the performance of the main-NLU which uses the same 62 intents. Also, since we do not have manual testing data for all 2,658 intents included in AZ, the performance of the AZ-62 on the Manual-62 test set is a good representative for how well any randomly chosen 62 intents of the backup-NLU might perform. There is no overlapping training data (by means of utterances) between the AZ and the Manual-62 test set.

Generated: LLM-Generated Training Data. In addition to sentences, we asked the LLaMA 3 model to generate a summary and keywords for each page. The three components (sentences from the summary, questions, and keywords) form an additional potential training set for the backup-NLU. The performance of this set is only evaluated on the Manual-62 test set.

5 Experiments, Evaluation, and Results

To assess the backup-NLU’s performance in detail and evaluate the effect of its features, we evaluated it across seven experimental conditions, each testing a different hypothesis regarding data source, structure, or pipeline configuration. We primarily used Top-N Accuracy, as our deployed chatbot aims to return multiple intent matches. Macro-averaged precision, recall, and F1 were also computed for

¹ Term Frequency - Inverse Document Frequency.

Top-1 predictions but did not yield surprising results on top of the accuracy score. An overview of all experiments, the features, and the test set used can be found in Table 3.

5.1 Feature Value Comparison

Figure 2a compares models trained on individual features (titles, keywords, and sentences) and their combination using the Manual-62 test set. Sentences alone provide the highest accuracy, keywords are the least effective, and combining all features yields the best performance.

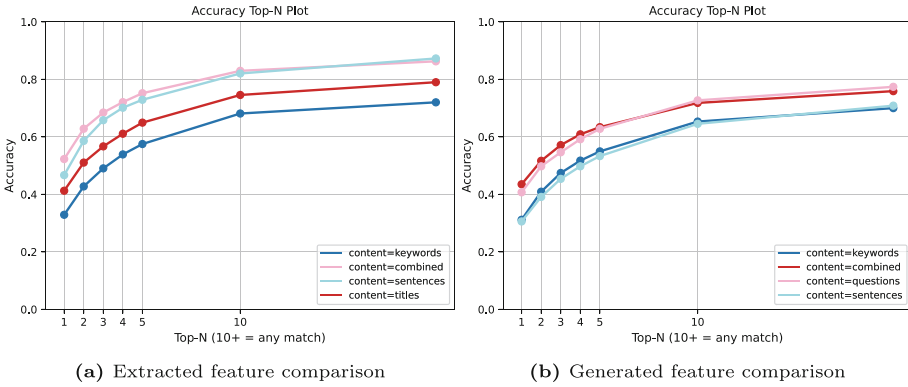


Fig. 2. Accuracy Top-N Plot for feature combinations on the Manual-62 test set.

5.2 Generated Feature Experiment

Figure 2b shows results using LLM-generated keywords, questions, summaries (i.e., sentences), and their combination. Interestingly, questions alone performed nearly as well as the full combination and even better at Top-5 to Top-10 levels. This could be due to the structural similarity to the evaluation utterances. Using sentences and keywords only yielded similar results.

5.3 Stripping (Pipeline) Experiment

Figure 3a analyzes whether stripping utterances and training data down to the contained keywords improves performance. Interestingly, stripped training data consistently underperforms, even when using stripped input. The best results come from processing original utterances via the pipeline method that strips the original input utterance (from the test set) down into its keywords and sends both the original and the keyword-utterance to the NLU. The intent ranking of both results is combined. We are using this testing method in further experiments.

Example keyword-utterance: “How do I enroll into a course at the TU Berlin after admission?” → “enroll course TU Berlin after admission”.

Table 3. A summary of each experiment, its training features, and test set.

Features	Input	Dataset
Extracted Features Comparison		
This experiment compares the value of different extracted features for training.		
combined		Manual-62
sentences		
titles		
keywords		
Generated Features Comparison		
Similar to the feature experiment, but using LLM-generated content.		
combined		Manual-62
sentences		
questions		
keywords		
Training Data Source		
Comparing extracted features only to LLM-generated features.		
combined		Manual-62
extracted		
generated		
Pipeline / Stripping Experiment		
Comparing whether stripping long utterances and training data down to essential keywords improves performance.		
combined	utterance	Manual-62
original	keyword-utterance	
stripped	both / pipeline	
Test-set-Experiment		
Comparing different test sets.		
AZ		Manual-62
AZ-62		LLM-62
		LLM
Depth Level - Experiment		
Comparing training the backup-NLU on slices of the page tree grouped by depth.		
depth X		LLM-X
NLU-Experiment		
main-NLU vs. backup-NLU (AZ)		
main-NLU		Manual-62
AZ-62		LLM-X
AZ		LLM-62

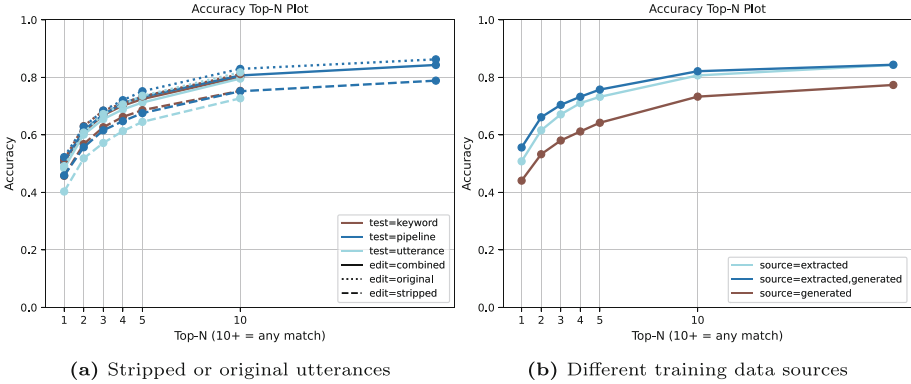


Fig. 3. Accuracy Top-N Plots on the manual test set for the stripping experiment and the training data source comparison.

5.4 Training Data Source Comparison

In Fig. 3b, we compare models trained on extracted data, LLM-generated data, and a combination of both. Models trained solely on extracted data outperform those using LLM-generated data. The combination performs best, showing complementary effects. In further experiments of this paper, only the extracted data is used for training to allow the LLM-generated data to serve as the second test set.

5.5 Test Set Comparison

Figure 4a compares the performance of backup-NLU on the test set. The backup-NLU model that only contains the 62 intents (AZ-62) performed best on the LLM-62 test set and moderately on the Manual-62 test set. The complete AZ model, however, performed best on the LLM-full data set, which contains all intents. When testing only the 62 intents (LLM-62 or Manual-62), the performance is worse, likely due to the confusion of similar intents that the test set does not consider. Overall, there is a significant performance drop when introducing all 2,658 intents vs only using the selected 62.

5.6 Depth-Based Training Analysis

Figure 4b examines how training data from different hierarchical depths affect accuracy. For this experiment we are using the LLM-X test set and match the testing intents to the training intents. The higher the depth, the more intents are in the training and test set, meaning X is growing with growing depth (Depth-0: LLM-10, depth-1: LLM-114, ..., Depth-9: LLM-2449, AZ: LLM-2658.) Performance declines with deeper levels, plateauing at a depth of around 6.

While using fewer topics increases performance, a cut-off prevents direct mapping to topics of the lower depth levels. This experiment was conducted to check

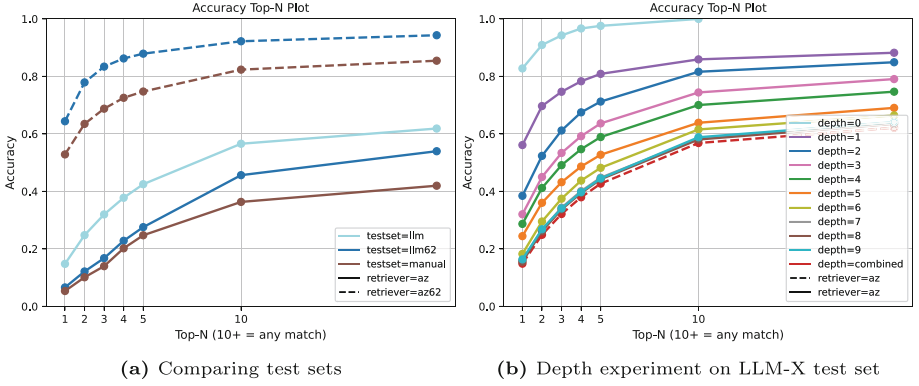


Fig. 4. Accuracy Top-N Plots for a comparison of the performance of the backup-NLU model on different test sets and using a different amount of topics (based on tree depth). a) Backup-NLU models (AZ and AZ62) on different test sets. b) Depth performance comparison. For each depth, the test set uses the same intents as the training set.

if performance could be increased by focusing on the most important pages. All other pages are still reachable through conversation with the chatbot due to the tree structure. The result suggests either a very high cut-off or none at all.

5.7 Main-NLU Vs Backup-NLU

Finally, Fig. 5 shows the main-NLU’s advantage on its own training data (Manual-62). However, the backup-NLU (AZ-62) outperforms it on the synthetic LLM-62 test set. It even performs better on the Manual-62 (the more challenging test set) than the main-NLU performs on the LLM-62 (the easier test set).

6 Discussion

Our experiments reveal that while there is room for improvement, scalable intent classification is viable using only extracted training data. While manually labeled data can optimize performance on known topics, backup-NLU models offer broader generalization. Combining a vast amount of features maximized performance, while generated data is not generally more valuable than directly extracted data. Further, using only the upper depths for training and, therefore, using less intents overall could significantly increase performance. However, while stripping the input utterances, i.e., the text to be classified, down to just the keywords does offer some value, stripping the training data down can even decrease performance. Surprisingly, the results suggest that our manual training data set provides no additional value besides being limited to fewer intents and, therefore, performing more accurately. Creating smaller data sets can be easily done with the generated data (e.g., AZ-62). Combining Manual-62 and AZ-62 into one data set would be an interesting experiment to do next. Outsourcing the

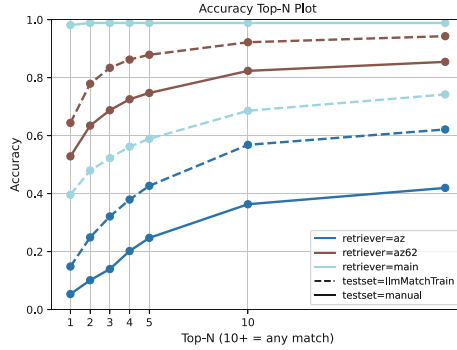


Fig. 5. Top-N Accuracy for main and backup-NLU variants on the manual test set (Manual-62) and the LLM test set (LLM-X/llmMatchTrain matches the testing intents to the training intents).

most important intents into a main-NLU makes sense and benefits the system’s performance.

Comparing the main-NLU and the backup-NLU would not have been possible without a new, automatically generated data set. Despite some noise in LLM-generated content, such datasets remain useful for relative performance comparison.

6.1 Reproducibility and Limitations

This study was conducted on a real-world deployment with secure infrastructure running on our own servers, including a custom-built website scraper and a two-step NLU pipeline. Full reproducibility of this study requires access to the underlying data hierarchy and tree structure of the TU Berlin website in the exact state used during training.

Additionally, while synthetic test sets (e.g., LLM-62, LLM-Full) provide broad coverage and can show trends, they may overestimate real-world performance due to alignment with LLM-generated phrasing. Similarly, the manual test set reflects only a narrow domain slice (62 intents out of 2.568) and is biased toward the main-NLU’s scope.

Another constraint lies in evaluation: current metrics treat each intent as a flat label despite the semantic overlap between many topics. Misclassifications may reflect ambiguous phrasing or multiple valid mappings. While Top-N accuracy partly compensates for this, future work could adopt intent clustering or use semantic-aware evaluation methods.

Finally, even though the system is mainly self-maintaining, its quality remains sensitive to the configuration of the web scraper. The filtering depth and pruning heuristics make a significant difference in performance. Future general-purpose applications may require domain-specific tuning to avoid overfitting or over-generation.

7 Future Work

This study opens multiple directions for follow-up exploration in order to optimize the performance of an automatically trained intent classification.

Hybrid Integration with RAG: Use the top-N suggestions from the backup-NLU as candidate retrieval sources in a Retrieval-Augmented Generation (RAG) pipeline, enabling more natural and flexible answers beyond static intent linking. We are already using this in our system but would like to run an in-depth analysis on this as well.

Cluster-Aware Evaluation: Since many topics are semantically similar or nested (e.g., different examination regulations), evaluation should tolerate near-miss predictions and use intent clustering or hierarchy-aware metrics.

User Logs: We are currently looking into incorporating real chatbot usage data from the last years to validate or refine our models.

Scalable Pruning: Instead of simple heuristics (e.g., too many descendants at a deep depth level), we would like to look into dynamically determining which parts of the scraped tree hierarchy should be included in training and which should remain navigational-only, based on performance-to-coverage trade-offs.

8 Conclusion

This paper presents a self-maintaining intent classification system trained entirely on structured web data and deployed as a backup-NLU for a university chatbot. Our approach removes the need for extensive manual annotation and scales to thousands of dynamic topics while preserving reasonable classification accuracy. We demonstrate that automated training pipelines can match or exceed traditional manually built NLUs in generalization settings through extensive experiments comparing training sources, feature sets, and test conditions.

These results demonstrate that automatic training pipelines can serve as reliable, general-purpose NLU modules in website-based environments with a high number of dynamically changing intents and pages. These lower-cost models may outperform manually trained and curated models, especially regarding adaption speed and effort.

Acknowledgments. Parts of the presented work and this paper have been funded by the Federal Ministry of Research, Technology and Space (Germany) and the Federal State of Berlin under grant no. 16DHBKI088 for the project USOS at Technische Universität Berlin.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Abdellatif, A., Badran, K., Costa, D.E., Shihab, E.: A transformer-based approach for augmenting software engineering chatbots datasets. In: Proceedings of the 18th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, ESEM 2024, pp. 359–370. Association for Computing Machinery, New York (2024). <https://doi.org/10.1145/3674805.3686695>
2. Ahmad, A., Kowol, P., Hillmann, S., Möller, S.: Multi-intent recognition in dialogue understanding: a comparison between smaller open-source LLMs. In: Proceedings of the 14th International Workshop on Spoken Dialogue Systems Technology, Sapporo, Japan (2024)
3. Benayas, A., Miguel-Ángel, S., Mora-Cantalops, M.: Enhancing intent classifier training with large language model-generated data. *Appl. Artif. Intell.* **38**(1), 2414483 (2024). <https://doi.org/10.1080/08839514.2024.2414483>
4. Bunk, T., Varshneya, D., Vlasov, V., Nichol, A.: DIET: lightweight language understanding for dialogue systems (2020). <https://doi.org/10.48550/ARXIV.2004.09936>
5. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic BERT sentence embedding. In: Proceedings of the 60th Annual Meeting of ACL, pp. 878–891. ACL, Dublin (2022). <https://doi.org/10.18653/v1/2022.acl-long.62>
6. Hillmann, S., Görzig, P., Möller, S.: Automatic generation of website-based multi-turn question-answering dialog systems. In: *Elektronische Sprachsignalverarbeitung 2023*, vol. 105, pp. 48–55. TUDpress (2023)
7. Hillmann, S., Kowol, P., Ahmad, A., Tang, R., Möller, S.: Usability and user experience of a chatbot for student support, pp. 22–29. TUDpress (2024)
8. Inupakutika, D., Akopian, D., Reddy, G., Chalela, P., Kaghyan, S., Mundlamuri, R.: Custom natural language understanding for healthcare chatbots and a case study. In: 2024 IEEE International Conference on Digital Health (ICDH), pp. 114–122 (2024). <https://doi.org/10.1109/ICDH62654.2024.00028>



Enhancing Retrieval Performance via LLM Hard-Negative Filtering

Danil Tirskikh¹ , Olesia Koroteeva¹ , Yuri Matveev¹ ,
Ekaterina Brovkina¹ , and Larisa Gonchar²

¹ ITMO University, Saint Petersburg 197101, Russian Federation
`danil.tirskikh@niuitmo.ru`, `{makhnytkina,yunmatveev}@itmo.ru`

² Saint Petersburg Mining University, Saint Petersburg 199106, Russian Federation

Abstract. In recent years contrastive learning has become the prevalent approach to training embedding models used in tasks such as information retrieval. Although, due to the nature of contrastive learning, during which negative samples are pushed further apart and positive samples are brought closer together in the embedding space, it imposed several new challenges for effective training. One such challenge lies in the creation of adequately hard negative samples. Most commonly hard-negative samples are created automatically by leveraging an existing retrieval model, which in turn introduces the risk of generating false negative training samples. To combat that, several filtering approaches that rely on ranks or similarity scores have been proposed. The fundamental flaw in those approaches lies in the ambiguity of relevance judgment given by the ranking models. Given only similarity scores, it is hard to accurately determine whether a particular document constitutes a positive or a negative sample. Our proposed method uses LLMs to resolve that issue by determining an optimal cutoff position through generating binary relevance labels. Such an approach also largely mitigates the problems with determining optimal filtration parameters present when dealing with raw relevance scores. The effectiveness of our method has been tested empirically by fine-tuning open-source retrieval models BGE-Reranker-v2-m3 and multilingual-e5-base. Our experiments on publicly available datasets have shown improvements in ranking metrics up to 2,29% in R-Precision and 1,12% in NDCG@10 compared to existing approaches.

Keywords: Information retrieval · LLM · Negative sampling

1 Introduction

Information retrieval for textual data encompasses a variety of methods ranging from full-text search algorithms such as BM25 to modern retrieval models based on neural networks that work with dense embeddings. Unlike full-text algorithms, neural network based approaches require extensive training to be useful in real world scenarios and are often tuned for a specific retrieval task

or several tasks in case of multi-task learning [1]. Such training is most often conducted following the contrastive learning paradigm. The main idea of contrastive learning is to bring the so called anchor embedding closer to a positive sample in the vector space, while simultaneously pushing negative samples further away [2]. In the case of textual retrieval, the anchor is the search query and relevant and irrelevant documents correspond to the positive and negative samples, respectively. This reliance on both negative and positive samples during training allows for better distinguishing ability of retrieval models which leads to significant performance gains.

But contrastive learning also imposes some challenges. Due to the active role of negative samples during the training process the performance of the resulting model depends largely on their quality and difficulty. Harder negatives lead to better performance by teaching the model to pay attention to finer semantic differences in texts. Unfortunately most publicly available datasets used for training lack explicit negative samples annotated by humans or otherwise created synthetically. Most of the time hard-negative samples are generated automatically by leveraging an existing pretrained retrieval model. By selecting the top-k most semantically similar documents for a given query as negative samples, excluding the ones that are labeled as relevant, we can ensure their difficulty.

Unfortunately this naive approach can lead to a large portion of false negative samples penetrating into the training process. As with a sufficiently large collection of documents we retrieve from, the probability to encounter an unlabeled relevant document for a particular query rises significantly. False negative samples adversely affect training and lead to suboptimal performance of retrieval models. Fine-tuning on such data can even cause performance degradation compared to the base model.

To combat the effects of false negatives various filtration methods have been developed by researchers over the years. Most common approaches rely on similarity scores produced by retrieval models to set up a threshold that would filter out any possible false negative samples. Similarly filtration can also be done based on the ranks of documents obtained by sorting them by their relevance scores. These approaches suffer from one fundamental flaw - scores and ranks show the order of relevance but say nothing of the relevance itself. It lead to several challenges, including how to accurately determine optimal filtration parameters without knowing the true relevance of each document. This problem is further exasperated by the fact that the distributions of similarity scores produced by different retrieval models are usually not calibrated and vary greatly from model to model [3].

Given the issues with existing methods, the goal was to create a new approach that avoids those drawbacks. The key idea was to move away from using relevance scores from retrieval models to identify false negatives. Instead, the method we propose relies on large language models (LLMs) to automatically generate binary relevance labels, that are then used to determine the optimal cutoff position among ranked negative documents. We call this new approach LLM-Cutoff.

2 Related Work

2.1 Negative Samples Mining

Various methods of automatically generating negative samples have been proposed throughout the years. Among them the following types can be distinguished. Random negative samples – this approach uses randomly selected documents from a given corpus as negative examples, provided they are not labeled as relevant to the given query [4].

In-batch negatives – at each training step, negative samples are formed within a specific batch, where the relevant documents for all other queries in the batch serve as negative samples for the current query [5].

Hard-Negative samples – these are generated either by the model being trained or by an external teacher model, by selecting the top-ranked documents for a given query from those not labeled as relevant [6].

The main problem with random negative samples is that most of them tend to be relatively easy. When sampling randomly from a large document collection, the likelihood of selecting negative samples that are semantically close to the relevant document is quite low. As a result, the model can easily learn to handle such negative samples because they differ significantly from relevant documents.

However, since the goal of retrieval models is to rank documents in terms of their relevance to a specific query, the documents appearing at the top of the retrieval model’s output are often semantically similar to each other. Therefore, to achieve high search accuracy, the model must be capable of detecting subtle differences between semantically similar documents, thereby ensuring the selection of the most relevant candidate.

2.2 Hard-Negative Filtering

To mitigate the impact of false-negative documents, a variety of approaches have been developed. For example, the work “Debiased Contrastive Learning” introduced a specialized loss function that accounts for the presence of false-negative samples in the training dataset [7]. However, this approach primarily aims to reduce the effects of false negatives rather than directly eliminating them.

Another line of research focuses on pre-filtering false-negative samples. This approach provides more flexibility in choosing the loss function and improves the overall quality of the training data. In the “RocketQA” study [8], filtering was performed using an absolute threshold. Specifically, the top-K samples retrieved by a retriever model were re-ranked by a cross-encoder trained on these samples. All instances with a relevance score above 0.1 were then discarded. The remaining samples were used for further training of the retriever model. A significant drawback of this method is the need to train a separate model specifically for filtering negative examples.

In the “SimANS” study [9], negative samples were selected based on a statistical distribution that favored those with relevance scores close to that of

positive documents. This allowed the method to exclude both highly irrelevant negatives—comparable in difficulty to random samples—and those with overly high relevance scores, which are more likely to be false negatives.

This idea was further developed in “TriSampler” [10], where the sampling distribution for negative samples was also based on relevance scores between the positive document and the negative documents. This added additional constraints to the sampling space and increased the informativeness of the selected negatives.

In the study “NV-Retriever: Improving Text Embedding Models with Effective Hard-Negative Mining”, the following relevance score based filtering methods were investigated [11]: TopK-Abs – absolute threshold, as defined in Eq. (1); TopK-Shifted – shifting the rank of the first hard-negative sample by k positions; TopK-MarginPos – threshold shifted relative to the relevance score of the positive pair by a fixed absolute value, as defined in Eq. (2); TopK-PercPos – threshold defined as a percentage of the relevance score of the positive pair, as defined in Eq. (3).

$$score_n < a \quad (1)$$

$$score_n < score_p - a \quad (2)$$

$$score_n < score_p * a \quad (3)$$

The subscripts n and p correspond to negative and positive scores respectively and a is a filtration hyperparameter. The aforementioned methods will form a baseline for our experiments.

The main drawback of such filtering methods is the difficulty of selecting an optimal threshold. It is often determined empirically by testing a range of possible values. This, in turn, significantly increases the number of experiments required and, consequently, the time and computational resources needed. Another challenge in determining the optimal filtering threshold for hard-negative samples is the fact that their true relevance is unknown. The relevance scores produced by retrieval models are continuous values, making it nearly impossible to identify a clear cutoff point between relevant and non-relevant documents. The relevance scores typically decrease gradually as the document rank decreases in the ranked list, without any obvious inflection points.

3 Proposed Method

Reflecting on the previously described issues with existing methods, the goal was to develop a new approach that avoids these shortcomings. The key step in creating such a method was to move away from using relevance scores from ranking models as the basis for identifying false-negative examples. Theoretically, if one can accurately determine the binary relevance of each document in the ranked list, it becomes possible to identify a rank threshold below which the majority of documents can be considered non-relevant with high confidence. To achieve this, the approach was shifted toward automatically generating binary

relevance labels using large language models (LLMs). The resulting method is referred to as LLM-Cutoff.

The core idea of the proposed method can be described through the following sequence of steps. First, hard-negative samples are obtained using an already trained retrieval model by selecting the top-k non-relevant documents for each query. Then, starting with a predefined step size (denoted as *step*), each document is checked for relevance using an LLM.

If a relevant document is found, the algorithm skips ahead by *step* ranks and continues checking. If a non-relevant document is encountered, the step size is reduced to 1, and documents are then checked one by one. This process continues until either a specified number of consecutive non-relevant documents is found, or the end of the list is reached. If a relevant document is encountered along the way, the step size is reset to *step*, and the process resumes.

The rationale behind the method is as follows: since we are working with a ranked list of documents, we assume that relevance tends to decrease with increasing rank. Therefore, there exists a point in the list after which all remaining documents can be considered non-relevant. As a result, it is unnecessary to check every document individually—we can use a step-based approach to reach this cutoff more efficiently. However, because neither the ranking model nor the LLM predictions are perfect, the threshold is not set based on the first non-relevant document found. Instead, we require a sequence of several consecutive non-relevant documents to increase confidence in the chosen cutoff point. A more naive approach would be to sample all top document that were deemed not relevant by an LLM - we dub this method LLM-Pluck. It is expected that due to relevance labeling errors made by the LLM, as well as ranking errors made by the retrieval model, this method will perform worse.

4 Experimental Setup

4.1 Datasets

All experiments were conducted on Russian language datasets. Out of all the datasets considered, the *medotvet-questions* dataset met all the necessary criteria. It consists of questions submitted by users on a medical online forum and responses provided by medical professionals. Additionally, the *rus_med_dialogues_qa* dataset was used, where responses to medical complaints were generated synthetically using an LLM. Since user complaints can be similar and doctors' responses may be relevant to multiple cases, the likelihood of false-negative examples in such data is particularly high. Therefore, these datasets are well-suited for evaluating the effectiveness of the proposed methods. The characteristics of the utilized datasets are presented in Table 1.

4.2 Models and Training Parameters

The Multilingual-E5-Large model was chosen as the retrieval model for computing relevance scores used in the creation of hard-negative samples. For every

Table 1. Dataset characteristics.

Split	medotvet-questions	rus_med_dialogues_qa
Train size	3839	2960
Validation size	216	389
Test size	216	393
Mean query length (words)	55	17
Mean candidate length (words)	40	63

query in each of the two selected datasets 15 hard-negatives samples were generated following each filtration method. For a more thorough assessment, two models were trained to evaluate the effectiveness of each approach: Multilingual-E5-Base and BGE-Reranker-v2-m3. Both models were trained as Cross-Encoder models. The infoNCE loss function [12] was used, and training was conducted for 5 epochs with an effective batch size of 16 on one NVIDIA A100 GPU. Two different LLMs were used to conduct our experiments - proprietary GigaChat-Pro model and an open-source Vikhr-Qwen-2.5-1.5B-Instruct model. These models were used because they were trained specifically to work well with the Russian language.

4.3 Prompting

A generic prompt was used to get relevance labels from each of the large language models used in our experiments. The English translated version of the prompt goes as follows (the original prompt was composed in Russian) - 'Determine whether the following document is relevant to the search query. Write "relevant" or "not relevant" depending on the answer. Do not repeat the content of the query or the document.'. The query and one of its corresponding relevant documents are also provided in each prompt as an example of true relevance. The goal of an LLM is to predict the relevance of an unlabeled document and only output a label without providing any explanations.

The same generic prompt was used for each model and dataset to test the generalizability of the proposed methods. However, tailoring the prompt to a specific LLM and dataset could increase the overall accuracy of relevance predictions and provide better filtering results, therefore increasing the final retrieval metrics. More advanced prompting techniques such as chain-of-thought prompting and the use of reasoning in LLM models could also provide a boost to LLM judgment quality, albeit at the cost of inference time.

Effectively fine-tuning the prompt for a specific case would require human validation of LLM relevance labels and would consequently introduce additional labor costs for the retrieval model training process. Therefore, in our work the quality of LLM relevance labels is assessed indirectly using the final retrieval metrics of the models tuned on generated data.

5 Results

5.1 Baseline Methods Comparison

For evaluating the performance of retrieval models standard information retrieval metrics NDCG@10 and R-precision were used. Let us now move on to a full comparison of the baseline methods on the previously mentioned datasets and models. The evaluation metrics obtained for the Multilingual-E5-Base model are presented in the Table 2.

Table 2. Baseline filtration performance for Multilingual-E5-Base.

Model	Multilingual-E5-Base			
Dataset	Medotvet-questions		Rus_med_dialogues_qa	
Metrics	NDCG@10	R-precision	NDCG@10	R-precision
Naive*	15.89	5.15	18.37	4.83
TopK-Shifted	21.07 (+5.18)	6.58 (+1.43)	24.76 (+6.39)	8.14 (+3.31)
TopK-Abs	28.31 (+12.42)	9.90 (+4.75)	25.73 (+7.36)	6.87 (+2.04)
TopK-MarginPos	29.81 (+13.92)	10.18 (+5.03)	26.16 (+7.79)	6.87 (+2.04)
TopK-PercPos	30.87 (+14.98)	8.71 (+3.56)	25.95 (+7.58)	7.38 (+2.55)

The *Naive* method, which involves no filtering of hard-negative samples, showed the worst results across all metrics. This supports the hypothesis that false-negative samples have a significant detrimental impact on the training process. Applying filtering led to significant metric improvements—with up to a 15% increase in NDCG@10 for the medotvet-questions dataset and up to 7.6% for rus_med_dialogues_qa. To verify the consistency of these improvements, the same set of experiments was conducted using the BGE-Reranker-v2-m3 model. The metrics obtained for this model are presented in Table 3.

Table 3. Baseline filtration performance for BGE-Reranker-v2-m3.

Model	BGE-Reranker-v2-m3			
Dataset	Medotvet-questions		Rus_med_dialogues_qa	
Metrics	NDCG@10	R-precision	NDCG@10	R-precision
Naive*	27.6	10.3	25.54	7.12
TopK-Shifted	30.71 (+3.11)	11.43 (+1.13)	26.8 (+1.26)	8.4 (+1.28)
TopK-Abs	34.65 (+7.05)	13.72 (+3.42)	27.93 (+2.39)	8.4 (+1.28)
TopK-MarginPos	35.37 (+7.77)	12.63 (+2.33)	29.21 (+3.67)	10.05 (+2.93)
TopK-PercPos	35.65 (+8.05)	12.74 (+2.44)	28.66 (+3.12)	9.16 (+2.04)

The conclusions drawn earlier also hold true for the BGE-Reranker-v2-m3 model. Applying hard-negative filtering methods helps eliminate false negatives and consistently improves performance. Since this model is several times larger than Multilingual-E5-Base and more thoroughly pre-trained, the absolute gains achieved through filtering are somewhat smaller. Nevertheless, the improvements remain significant—with up to 8% on the medotvet-questions dataset for the NDCG@10 metric, and up to 3.6% on rus_med_dialogues_qa.

5.2 LLM-Cutoff and LLM-Pluck Comparison

Turning to the evaluation of the proposed methods for filtering negative training samples, let us begin with a comparison of the two methods: LLM-Cutoff and LLM-Pluck. As mentioned earlier, it was expected that the LLM-Cutoff method would demonstrate higher performance metrics compared to LLM-Pluck. This assumption was based on the fact that LLM-Cutoff operates by identifying a relevance threshold within a ranked list of documents. The accuracy of this threshold is supported by the requirement of encountering N consecutive negative documents, which helps minimize the impact of random errors made by the LLM. In contrast, LLM-Pluck is much more susceptible to annotation errors from the LLM, as it includes all documents the LLM considers negative without further filtering. The results of training the Multilingual-E5-Base model on datasets prepared using these methods are presented in Table 4.

Table 4. LLM-Cutoff and LLM-Pluck performance for Multilingual-E5-Base.

Model	Multilingual-E5-Base			
Dataset	Medotvet-questions		Rus_med_dialogues_qa	
Metrics	NDCG@10	R-precision	NDCG@10	R-precision
LLM-Pluck (G)*	26.44	10.39	26.14	8.91
LLM-Cutoff (G)	31.11 (+4.67)	12.37 (+1.98)	27.28 (+1.14)	9.67 (+0.76)
LLM-Pluck (Q)*	22.25	6.67	22.71	6.87
LLM-Cutoff (Q)	24.35 (+2.10)	6.79 (+0.12)	26.13 (+3.42)	7.12 (+0.25)

As evident from the experimental results, the earlier assumptions about the advantages of the LLM-Cutoff method proved to be correct. The difference compared to the LLM-Pluck method reached up to 4.6% in the NDCG@10 metric and up to 2% in R-Precision on the medotvet-questions dataset. Improvements were observed when using both the proprietary GigaChat-Pro model—denoted in the table as (G)—and the open-source Vikhr-Qwen-2.5-1.5B-Instruct model—denoted as (Q).

The use of a larger proprietary model resulted in a significant improvement compared to the smaller open-source model. This improvement was observed across all datasets and for all methods. Specifically, using the GigaChat-Pro

model with the LLM-Cutoff method increased the NDCG@10 metric on the medotvet-questions dataset by nearly 7%, from 24.35 to 31.11. The R-Precision metric also improved by almost 6%, from 6.79 to 12.37. Similar results were confirmed for the BGE-Reranker-v2-m3 model. The experimental results for which are presented in Table 5.

Table 5. LLM-Cutoff and LLM-Pluck performance for BGE-Reranker-v2-m3.

Model	BGE-Reranker-v2-m3			
Dataset	Medotvet-questions		Rus_med_dialogues_qa	
Metrics	NDCG@10	R-precision	NDCG@10	R-precision
LLM-Pluck (G)*	33.46	11.59	28.36	10.18
LLM-Cutoff (G)	35.90 (+2.44)	15.04 (+3.45)	30.10 (+1.74)	11.20 (+1.02)
LLM-Pluck (Q)*	30.62	11.52	27.54	7.38
LLM-Cutoff (Q)	34.18 (+3.56)	12.78 (+1.26)	29.93 (+2.39)	10.94 (+3.56)

Similarly to the experiments with baseline negative sample filtration methods, the BGE-Reranker-v2-m3 model demonstrated better metrics compared to the Multilingual-E5-Base model. The performance gains from using a larger LLM and the LLM-Cutoff method are consistently observed for this model as well. Thus, we can confidently conclude the advantage of the LLM-Cutoff method over LLM-Pluck. The benefits of using large proprietary LLMs are also clearly evident in this comparison.

Let’s move on to comparing the best proposed method, LLM-Cutoff, with the best-performing baseline methods. The resulting metric values are presented in Table 6.

Table 6. LLM-Cutoff and LLM-Pluck performance for BGE-Reranker-v2-m3.

Model	Multilingual-E5-Base			
Dataset	Medotvet-questions		Rus_med_dialogues_qa	
Metrics	NDCG@10	R-precision	NDCG@10	R-precision
Best baseline*	30.87	10.18	26.16	8.14
LLM-Cutoff (G)	31.11 (+0.24)	12.37 (+2.19)	27.28 (+1.12)	9.67 (+1.53)
Model	BGE-Reranker-v2-m3			
Best baseline*	35.65	13.72	29.21	10.05
LLM-Cutoff (G)	35.90 (+0.25)	15.04 (+1.32)	30.10 (+0.89)	11.20 (+1.15)

The LLM-Cutoff method demonstrated better metrics compared to all baseline methods. The most significant improvement can be observed in the R-precision metric, which indicates ranking accuracy. The increase reached up to

2.2% for the medotvet-questions dataset. The improvement is stable across all trained models and all utilized datasets. This allows us to assert the advantage of the proposed method with a sufficient level of confidence. On average, the increase across all metrics was within 1–2 percent. Among the advantages of the proposed method, we can highlight the quality of the resulting data, achieved through the accuracy of determining the filtering threshold for hard-negative samples and the relative simplicity and flexibility of the algorithm itself.

Among the obvious drawbacks, one should note the speed of operation. It significantly lags behind baseline methods since using the LLM-Cutoff method requires multiple calls to large language models. As a result, the method is not suitable for use with excessively large datasets. Its most reasonable application is on small, specialized datasets that are used at the fine-tuning stage of training retrieval models.

Another possible drawback is related to the sensitivity of large language models to changes in prompts. Consequently, when switching LLMs, it may be necessary to revise the wording and format of the prompt being used. In cases where proprietary LLMs are employed, changes in model versions may occur unnoticed by the end user. This can lead to unexpected changes in the quality of the model’s responses and may require adjustments to the prompt.

6 Conclusion

In this paper we explore hard-negative samples filtration methods and propose our own method LLM-Cutoff based on the identified shortcomings of existing filtration methods. LLM-Cutoff utilizes large language models (LLMs) to determine optimal cutoff position in a ranked list of possible negative candidates. Major difference with existing methods lies in departure from utilizing continuous relevance scores provided by retrieval models, generating binary relevance labels via prompting instead. The use of the proposed method led to an improvement of 2.19% in R-precision on the medotvet-questions dataset and 1.53% on the rus_med_dialogues_qa dataset. Overall, the average improvement in the performance of the search models across all experiments was 1–2% compared to the existing baseline methods.

One of our method’s most obvious drawbacks is the processing speed. It is significantly slower compared to baseline methods, since the LLM-Cutoff method requires multiple calls to large language models. As a result, the proposed method is not suitable for use with very large datasets. Its most reasonable application is on small, narrowly focused datasets used during the fine-tuning of retrieval model for specific tasks or domains. In such contexts, the proposed method can significantly improve the quality of training data by more accurately filtering out false-negative examples—which, in turn, leads to better performance of resulting models.

However, the method’s robustness and generalizability require further investigation, as well as its applicability to more domains and languages. Another area of further research may lie in utilizing better prompting techniques such as

chain-of-thought or using large language models with reasoning capabilities to get more accurate binary relevance labels.

References

1. Masliukhin, S.M., Posokhov, P.A., Skrylnikov, S.S., Makhnytkina, O.V., Ivanovskaia, T.I.: Prompt-based multi-task learning for robust text retrieval. *J. Sci. Tech. Inf. Technol. Mech. Opt.* **24**(6), 1016–1023 (2024)
2. Khosla, P., et al.: Supervised contrastive learning. *Adv. Neural. Inf. Process. Syst.* **33**, 18661–18673 (2020)
3. Sheng, X.-R., et al.: Joint optimization of ranking and calibration with contextualized hybrid model. In: *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4813–4822 (2023)
4. Karpukhin, V., et al.: Dense passage retrieval for open-domain question answering. In: *EMNLP (1)*, pp. 6769–6781 (2020)
5. Zhan, J., Mao, J., Liu, Y., Zhang, M., Ma, S.: RepBERT: contextualized text embeddings for first-stage retrieval. *arXiv preprint [arXiv:2006.15498](https://arxiv.org/abs/2006.15498)* (2020)
6. Zhan, J., Mao, J., Liu, Y., Guo, J., Zhang, M., Ma, S.: Optimizing dense retrieval model training with hard negatives. In: *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1503–1512 (2021)
7. Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A., Jegelka, S.: Debaised contrastive learning. *Adv. Neural. Inf. Process. Syst.* **33**, 8765–8775 (2020)
8. Qu, Y., et al.: RocketQA: an optimized training approach to dense passage retrieval for open-domain question answering. *arXiv preprint [arXiv:2010.08191](https://arxiv.org/abs/2010.08191)* (2020)
9. Zhou, K., et al.: SimANS: simple ambiguous negatives sampling for dense text retrieval. *arXiv preprint [arXiv:2210.11773](https://arxiv.org/abs/2210.11773)* (2022)
10. Yang, Z., Shao, Z., Dong, Y., Tang, J.: TriSampler: a better negative sampling principle for dense retrieval. In: *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 9269–9277 (2024)
11. Gabriel de Souza, P.M., Osmulski, R., Xu, M., Ak, R., Schifferer, B., Oldridge, E.: NV-retriever: improving text embedding models with effective hard-negative mining. *arXiv preprint [arXiv:2407.15831](https://arxiv.org/abs/2407.15831)* 1 (2024)
12. Izacard, G., et al.: Unsupervised dense information retrieval with contrastive learning. *arXiv preprint [arXiv:2112.09118](https://arxiv.org/abs/2112.09118)* (2021)



Sector-Wise Backpropagation for Low-Resource Text Classification in Deep Models

José Luis Vázquez Noguera¹ , Carlos U. Valdez² , Marvin M. Agüero² ,
Julio C. Mello³ , José D. Colbes³ , and Sebastián A. Grillo^{2,3}

¹ Computer Engineer Department, Universidad Americana, Asunción, Paraguay

² Facultad de Ciencias y Tecnología, Universidad Autónoma de Asunción,
Asunción, Paraguay

{cavaldez,sgrillo}@uaa.edu.py

³ Facultad Politécnica, Universidad Nacional de Asunción, Asunción, Paraguay

Abstract. This work presents a modular training approach for deep neural networks applied to text classification, introducing the concept of a *sector* as a trainable layer and its subsequent non-trainable layers up to the next trainable layer. Each sector is trained independently using auxiliary models that mimic the original network's output layer, allowing for training without full end-to-end backpropagation. The method was evaluated on 1D ConvNet, Transformer, and bidirectional LSTM architectures across five benchmark text classification data sets, using 1%, 5%, and 10% of the data for training. Results show that sector-wise modular training achieves comparable or even superior accuracy to traditional end-to-end training while reducing computational time, particularly in four out of five data sets. This approach offers a generalizable alternative to standard training methods in deep learning for text classification.

Keywords: Text classification · Natural language understanding · Deep learning

1 Introduction

Text classification is a basic task for natural language processing and text mining problems. More particularly, within the broader field of Natural Language Understanding (NLU). NLU is concerned with enabling computers to understand and process human language, and text classification is a key component of that process. Essentially, text classification consists of assigning a categorical output to a given text input. Machine learning is a widely used approach to text classification. However, classic machine learning models requires a lot of feature engineering and relies heavily on domain knowledge [18].

Neural network techniques emerged as an alternative to handcrafted approaches, using embedding models that represent text as vectors in continuous spaces. Early attempts performed poorly compared to classical methods. The

change came with deep-layer models trained adjusting more parameters, such as Word2vec [17], Embeddings from Language Models (ELMO) [16], and especially with the arrival of Transformer [25], which enabled significant improvements in efficiency and performance.

The introduction of these deep-layer architectures marked a turning point in NLP, leading in an era of pretrained models, that are adapted to downstream tasks via end-to-end fine-tuning. The development of this technology has mainly followed three paths: (i) layer-wise adaptation of pre-trained backbones, (ii) integrated low-resource and cross-lingual strategies, and (iii) lightweight architectures trained from scratch and optimised for rapid convergence [31].

Rather than updating an entire pre-trained model, recent work focuses on conditioning each layer independently. Discriminative fine-tuning uses separate learning rates per layer to stabilize low-level encoders and adapt higher ones [11]. Adapter modules insert small trainable layers while freezing the rest, achieving strong results with minimal updates [10]. BitFit goes further by only updating bias terms, showing $<0.1\%$ of weights can be enough [4].

In low-resource settings, researchers combine cross-lingual transfer, data augmentation, and self-training. Data set reconstruction creates synthetic examples from limited data, like Setswana headlines, for better generalization [15]. Self-training and prototypical meta-learning use unlabeled texts or learn prototypes across dialects to improve performance from few examples [21]. Continued masked pre-training on domain-specific text realigns multilingual models before fine-tuning [9].

Some efforts skip pre-training and design small transformer variants, like MobileBERT, TinyBERT, and SqueezeBERT, that train well from scratch using factorization and parameter sharing [12, 13, 24]. Pruning pretrained models, like reducing BERT layers for Guarani, also improves efficiency [1]. Tokenization-free models like CANINE operate on characters, benefiting rich-morphology languages [7]. Lightweight CNNs with strong regularization can also reach good accuracy in few epochs on benchmarks.

End-to-end backpropagation (E2EBP) has long been the standard for training deep networks by optimising all parameters simultaneously via forward and backward passes. However, it suffers from issues such as vanishing or exploding gradients and limited modularity, making debugging and interpretation difficult. To address these challenges, modular and weakly modular training approaches have been proposed. Modular training avoids end-to-end backward and forward passes, while weakly modular training requires only the forward pass [8]. A notable modular approach involves using auxiliary classifiers to guide feature learning. These classifiers are trained together with the feature extractor using a classification loss and discarded at test time. These types of models have been evaluated mainly on image data sets, with limited success [3].

This work proposes an application of modular training for deep neural networks, considering that not all layers contain trainable parameters (e.g., pooling layers). In this context, we adopt the notion of a “sector”, which consists of a parameterized layer and all subsequent non-trainable layers up to the next parameterized layer.

The applied modular training procedure involves: i) building new auxiliary models composed of a sector and an output layer similar to that of the original deep network, ii) training each of these auxiliary models using the output of the previous trained sector as input while preserving the original labels, and iii) transferring the trained parameters of each sector from the auxiliary models to the original deep network.

While transfer learning through pretrained large language models (such as BERT or RoBERTa [30]) has become a standard practice in NLP, it typically requires considerable computational resources and access to sufficient data for effective fine-tuning. In contrast, our sector-wise training paradigm is lightweight, model agnostic, and designed for scenarios where training must be efficient and feasible under tight constraints. Our method aims to improve generalization and optimization dynamics by partitioning a model into independently trained sectors (rather than relying on pretraining), each of which can learn localized representations. This modularity offers an alternative in low-resource settings where pretrained models are inaccessible or ineffective due to domain shift or limited adaptability.

This methodology was evaluated on architectures based on 1D ConvNet, Transformer, and bidirectional LSTM layers, across five benchmark text classification problems, using 1%, 5%, and 10% of the data set as training sets.

The main contributions of this work are:

1. The formulation of a modular training approach for deep networks that is generalizable to standard text classification architectures.
2. A comparison of sector-wise modular training in terms of accuracy and training time, against traditional E2EBP training (with and without Bayesian hyperparameter optimization). The results show that in 4 out of 5 data sets, sector-wise modular training outperformed the traditional methods in accuracy while achieving significant time savings.

The rest of this paper is organized as follows. Section 2 describes the proposed sector-wise modular training methodology. Section 3 presents the results in terms of accuracy and training time. Finally, Sect. 4 discusses the findings from the case studies.

2 Proposal

Traditional backpropagation only addresses the final error of the neural network, adjusting most parameters in each epoch. This can lead to overfitting, particularly when only limited data are available. To address this problem, our approach optimises the parameters of each layer separately rather than all at once. This limits the number of parameters being adjusted simultaneously, helping to prevent overfitting.

To this end, it is important to clarify that optimisation is carried out under the concept of ‘sectors’, where each sector consists of a parameterised layer and all other subsequent layers without parameters before reaching the next

parameterised layer. For example, suppose we want to train a C_1 - P_1 - P_2 - C_2 - P_3 - C_3 architecture where (i) from left to right we indicate the type of layer in ascending order in the network, (ii) C_i indicates a fully connected layer, and (iii) P_i indicates a pooling layer that lacks parameters to be adjusted. Therefore, the sectors that exist are $S_1 = C_1$ - P_1 - P_2 , $S_2 = C_2$ - P_3 and $S_3 = C_3$.

Let D denote a deep neural network and X the training set. The idea of the proposal consists of three steps:

1. For each sector S_i (excepting the last sector) of the deep network D construct a shallow network N_i consisting only of that sector and, on top of that, an output layer identical to the last layer of the deep network D .
2. Train each sector N_i using the instances $f_{i-1}(x)$ (where $f_0(x) = x$ and using the same label of x) and compute the instances $f_i(x)$ as the output of $f_{i-1}(x)$ when evaluated at N_i by removing the output layer.
3. Reconstruct the network D by taking the trained parameters of each N_i and removing the output layer (except for the last network N_i , which is incorporated as is).

Continuing with the previous example, we would have the networks $N_1 = C_1$ - P_1 - P_2 - C'_3 (where C'_3 is similar to C_3) and $N_2 = C_2$ - P_3 - C_3 . N_1 is trained using the instances $x \in X$, while N_2 is trained using the mapped instances $f_1(x)$. Finally, we discard C'_3 and its parameters from N_1 and connect it to N_2 , obtaining the same architecture D , but with trained parameters. Algorithm 1 presents the proposal by applying individual epochs until the stop condition is reached within the WHILE loop.

It should be noted that, a priori, it is not necessary to use regularization methods when training each auxiliary model N_i because it is a shallow architecture, even though it may have many non-parameterized layers. It can also be observed that the last layer of each auxiliary model is similar to that of the initial model based on two desirable properties: (i) that both models separate the same number of classes and (ii) maximize the linear separability of classes at each layer (which is motivated by the success of linear probes [2]).

From a theoretical point of view, the way in which auxiliary models are defined implies that model training tends to optimize the linear separability of classes. This is because auxiliary models are constructed by placing only one layer on top of the layer to be adjusted, so the output of that layer is adjusted to a linear transformation represented by the layer to be discarded. This particular implementation seeks to limit overfitting based on two strategies: i) training limits the number of parameters to be adjusted simultaneously, and ii) adjustment is gradual, using only one epoch per layer (which produces a gradual adjustment that is not as greedy as other proposals).

3 Results

The tests were applied to the following datasets: Large Movie Review [14], AG News [29], TREC-6 [26], SST-5 [23], and Glue [28]. For each dataset, a simple

Algorithm 1. Sector-wise Local Backpropagation and Network Reconstruction

```

1: Initialize: Architecture  $D$ , sectors  $S_1, \dots, S_n$ , null network  $R_0$ 
2: while not stop condition do
3:   Backpropagation in sectors
4:   for  $i = 1$  to  $n - 1$  do
5:     Create  $N_i$  by adding a layer similar to the last layer of  $D$  on top of sector  $S_i$ 
6:     Train  $N_i$  for one epoch using the instances  $f_{i-1}(x)$  for  $x \in X$ , with the same
       label as  $x$ 
7:     Calculate  $f_i(x)$  for each  $x \in X$ , evaluating  $f_{i-1}(x)$  for each  $x \in X$  according
       to the output of the penultimate layer of the trained network  $N_i$ 
8:   end for
9:   Network reconstruction
10:  Set  $R_0 \leftarrow$  empty network
11:  for  $i = 1$  to  $n - 2$  do
12:    Extract  $S_i$  from the trained  $N_i$  preserving learned parameters
13:    Connect the extracted  $S_i$  to  $R_{i-1}$  according to architecture  $D$ , forming  $R_i$ 
14:  end for
15:  Connect  $R_{n-2}$  to  $N_{n-1}$  according to architecture  $D$ , forming  $R_{i-1}$ 
16: end while
17: return  $R_{i-1}$ 

```

1D convnet architecture [20,27], a transformer-based architecture [19,25], and a bidirectional LSTM-based architecture [6,22] were applied. The methodology used to evaluate the algorithms for each $x\%$ of the dataset consisted of partitioning the dataset into an 80% training set and a 20% test set. A random $x\%$ of the dataset was then extracted from the training set and used to train the model, which was subsequently evaluated on the test set. This procedure was repeated 10 times to obtain an average error rate. The percentages chosen to apply this methodology were 1%, 5% and 10% in relation to the total dataset.

Each model applied to each dataset was tested using three possible training configurations. The first configuration is traditional backpropagation applied without hyperparameter search. The second configuration is backpropagation applied with Bayesian hyperparameter search. The third configuration consists of the proposal applying backpropagation by sectors. The first and third configurations of the algorithms were applied using the following parameters; batch size: 32, learning rate: 0.001, optimizer set to Adam, and loss set to sparse categorical cross-entropy. For the second configuration, Bayesian optimization was applied using the following domains for each hyperparameter: optimizer in ['adam', 'rmsprop', 'sgd'], learning rate in the range (1e-5, 1e-1) with a log-uniform distribution, and batch size in the range (16, 128).

Figures 1 and 2 show the accuracy percentage of each configuration and percentage for epochs 2 and 10, respectively, on the IMDB data set. Figure 3 presents the total time of these tests on the IMDB data set up to epoch 10. Figures 1 and 2 show that the proposed method achieved a lower error rate in all test cases. Figure 1 illustrates that the proposal reached faster convergence at epoch 2 for the Transformer architecture with 5% and 10% of the data, and for the

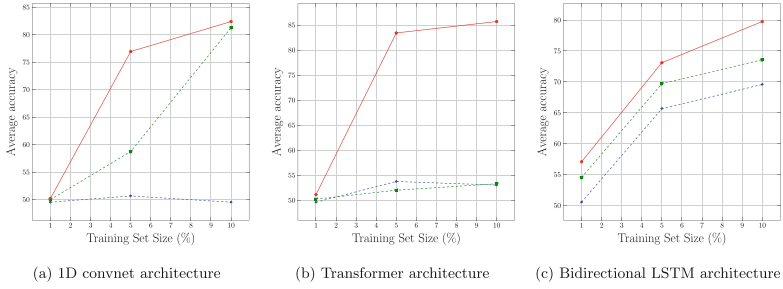


Fig. 1. Average accuracy of epoch 2 as a function of using 1%, 5% and 10% of the IMDB data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

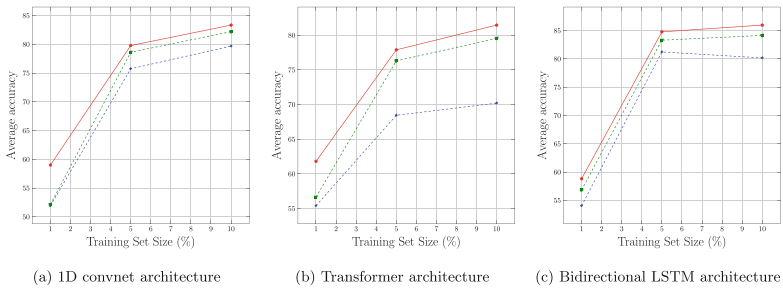


Fig. 2. Average accuracy of epoch 10 as a function of using 1%, 5% and 10% of the IMDB data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

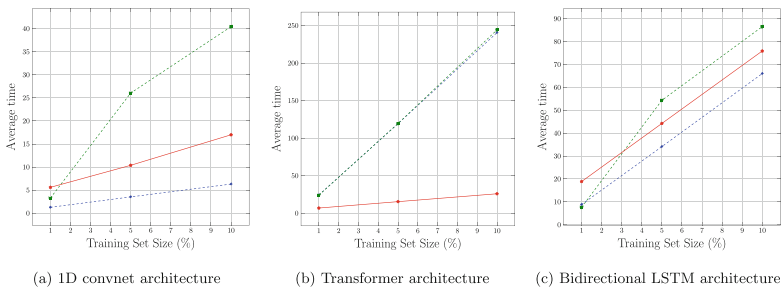


Fig. 3. Average time execution up to epoch 10 in seconds as a function of using 1%, 5% and 10% of the IMDB data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

1D ConvNet architecture with 5%. In contrast, Fig. 2 shows that at epoch 10, the proposed method presents a slight improvement in all cases. In terms of accuracy, the best configuration recorded is the Transformer-based architecture when trained with the proposal for 10 epochs. Figure 3 shows that the proposed method tends to gain relative computational efficiency as the data set size increases, as well as a significant time savings for the Transformer architecture.

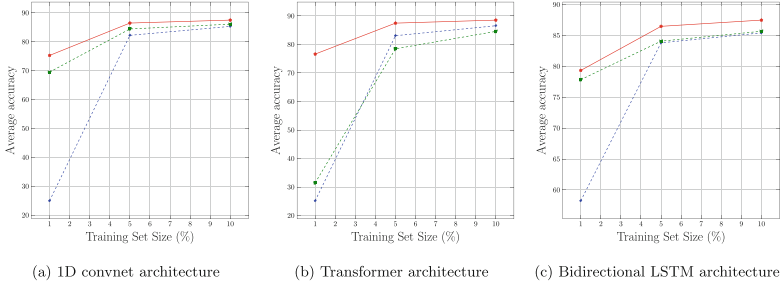


Fig. 4. Average accuracy of epoch 2 as a function of using 1%, 5% and 10% of the AG News data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

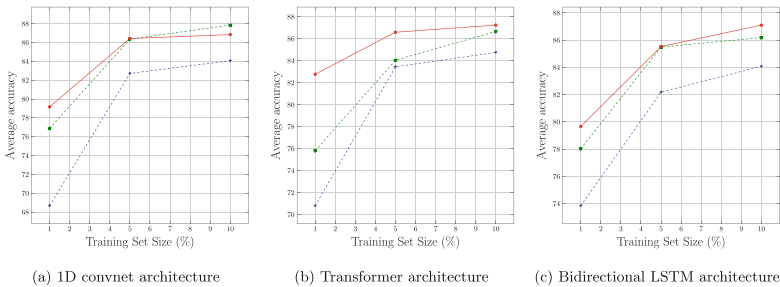


Fig. 5. Average accuracy of epoch 10 as a function of using 1%, 5% and 10% of the AG News data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

Figures 4 and 5 show the accuracy percentage of each configuration and percentage for epochs 2 and 10, respectively, on the AG News data set. Figure 6 presents the total time of these tests on the AG News data set up to epoch 10. At epoch 2, Fig. 4 shows a slight improvement in most cases for the proposed method, and significant improvements due to faster convergence for the Transformer architecture with 1% of the data set. Figure 5 shows a slight improvement

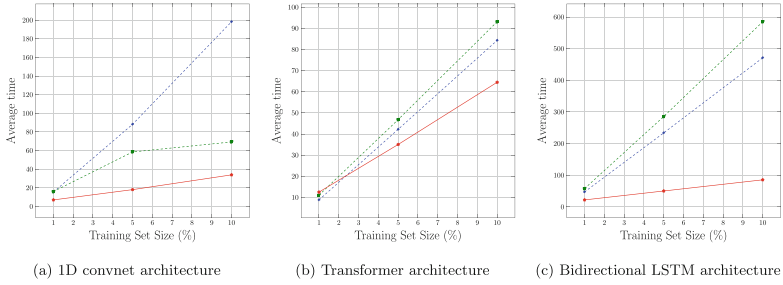


Fig. 6. Average time execution up to epoch 10 in seconds as a function of using 1%, 5% and 10% of the AG News data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

of the proposed method in almost all cases at epoch 10, except for the 1D ConvNet architecture, where it was slightly outperformed by the traditional method with Bayesian optimization using 10% of the data set. In terms of accuracy, the best configuration recorded is the Transformer-based architecture when trained with the proposal for 2 epochs. Figure 6 shows that the proposed method was more time-efficient at epoch 10 in all cases, except for the Transformer architecture with 1% of the data set, where the difference was minimal. For this data set, the proposed method achieved significant time savings for the Bidirectional LSTM architecture, followed by the 1D ConvNet architecture.

Figures 7 and 8 show the accuracy percentage of each configuration and percentage for epochs 2 and 10, respectively, on the TREC-6 data set. Figure 3 presents the total time of these tests on the TREC-6 data set up to epoch 10. Figure 7 shows that at epoch 2, the proposed method performed worse than all baseline models when using 1% and 5% as the training data. However, with 10% as the data, the proposed method outperformed the others. Figure 8, on the other hand, shows that the proposed method outperformed traditional techniques in most cases, except for the Transformer architecture with 1% as the training data. In terms of accuracy, the best configuration recorded is the architecture based on bidirectional LSTM when trained with the proposal for 10 epochs. Figure 9 shows that the proposed method resulted in significant time savings only for the bidirectional LSTM-based architecture when using 5% and 10% of the training data.

Figures 10 and 11 show the accuracy percentage of each configuration and percentage for epochs 2 and 10, respectively, on the SST-5 data set. Figure 3 presents the total time of these tests on the SST-5 data set up to epoch 10. Figure 10 shows mixed results for epoch 2, with some cases where the proposed method can be better or worse compared to traditional methods. However, the differences are relatively minor. Figure 11 shows that for epoch 10, the proposed method had lower error rates than traditional methods, except for the bidirectional LSTM architecture using 1% of the data set for training. In terms of

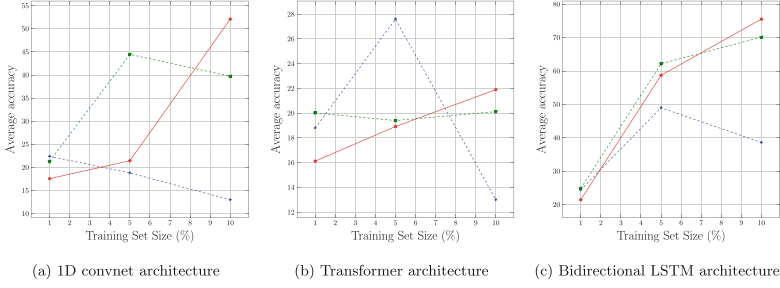


Fig. 7. Average accuracy of epoch 2 as a function of using 1%, 5% and 10% of the TREC-6 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

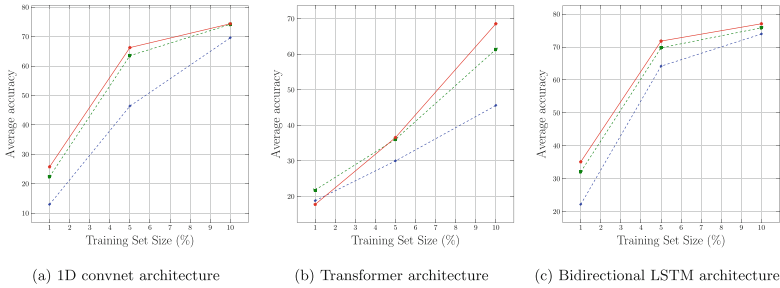


Fig. 8. Average accuracy of epoch 10 as a function of using 1%, 5% and 10% of the TREC-6 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

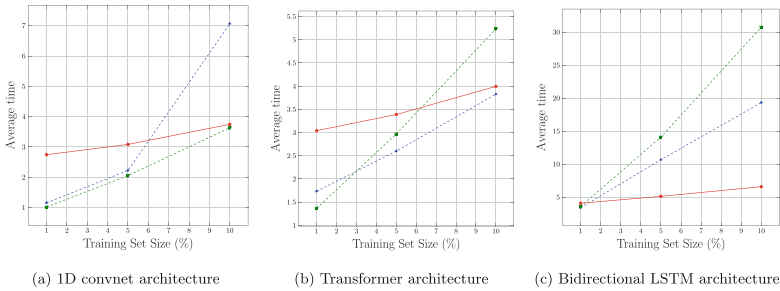


Fig. 9. Average time execution up to epoch 10 in seconds as a function of using 1%, 5% and 10% of the TREC-6 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

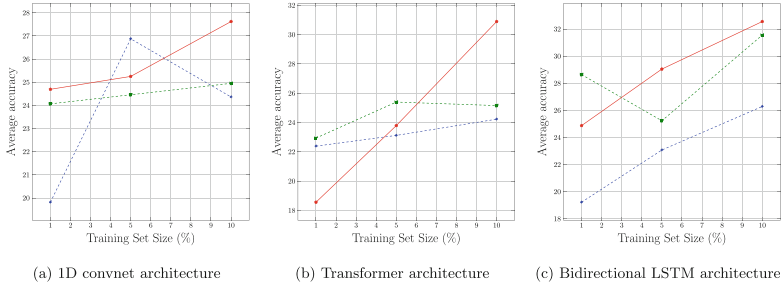


Fig. 10. Average accuracy of epoch 2 as a function of using 1%, 5% and 10% of the SST-5 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

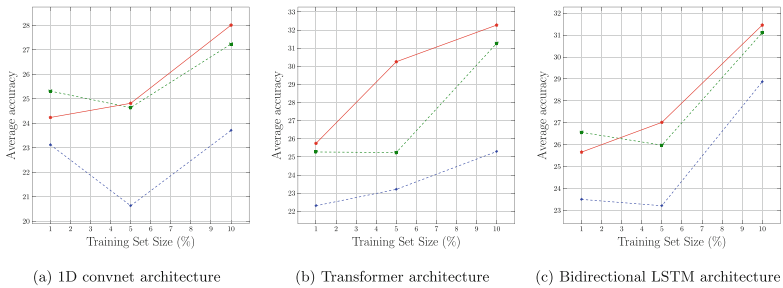


Fig. 11. Average accuracy of epoch 10 as a function of using 1%, 5% and 10% of the SST-5 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

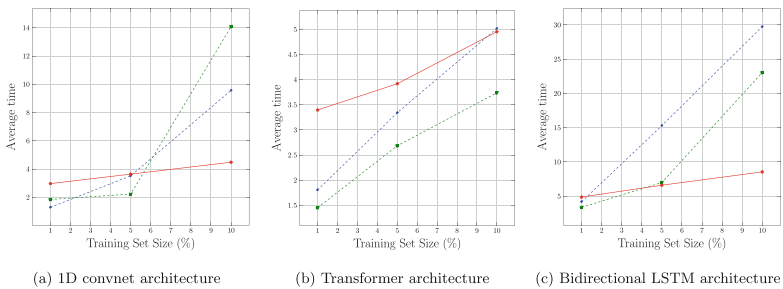


Fig. 12. Average time execution up to epoch 10 in seconds as a function of using 1%, 5% and 10% of the SST-5 data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

accuracy, the best configuration recorded was the architecture based on bidirectional LSTM when trained with the proposal for 2 epochs. Figure 12 illustrates a tendency of the proposed method to become more time-efficient as the training set size increases, but it only shows a significant advantage for the 1D ConvNet and bidirectional LSTM architectures when using 10% of the data set as the training set.

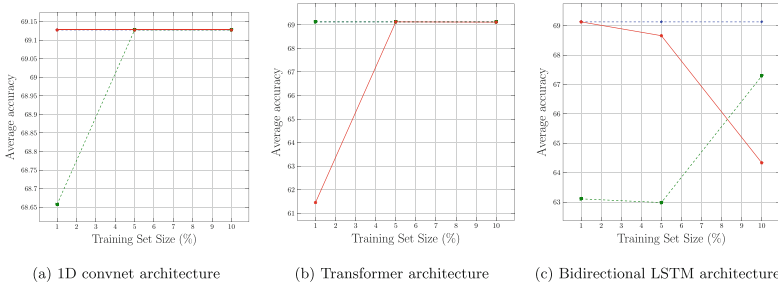


Fig. 13. Average accuracy of epoch 2 as a function of using 1%, 5% and 10% of the GLUE data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

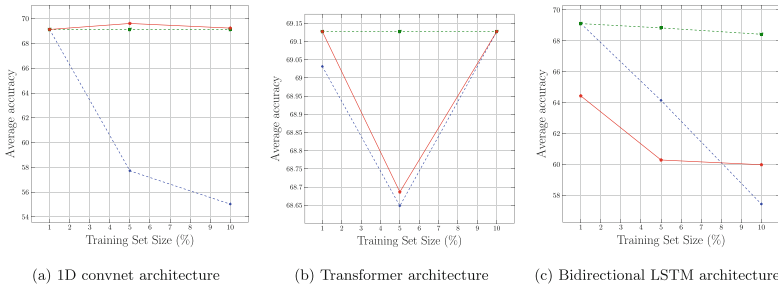


Fig. 14. Average accuracy of epoch 10 as a function of using 1%, 5% and 10% of the GLUE data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

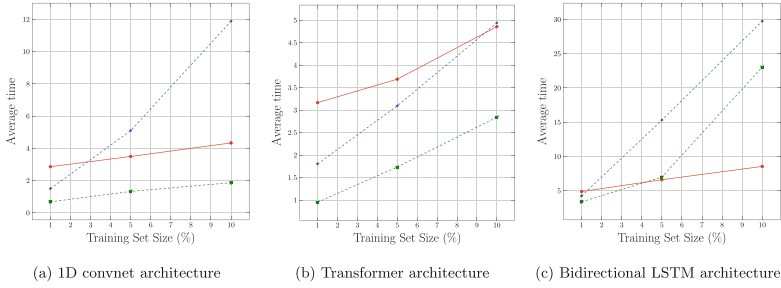


Fig. 15. Average time execution up to epoch 10 in seconds as a function of using 1%, 5% and 10% of the GLUE data set for training. The proposal is shown in red, traditional simple backpropagation in blue, and traditional backpropagation using Bayesian hyperparameter optimisation in green. (Color figure online)

Finally, Figs. 13 and 14 show the accuracy percentage of each configuration and percentage for epochs 2 and 10, respectively, on the Glue data set. Figure 3 presents the total time of these tests on the Glue data set up to epoch 10. Figure 13 shows that the proposal did not achieve any advantage for epoch 2 in terms of accuracy percentage, and the same can be seen in Fig. 14 for epoch 10. Figure 15 shows that the proposal only showed lower computational cost for the bidirectional LSTM-based architecture, however, the proposal had lower accuracy in that case than traditional methods and therefore the Glue data set was the only one where the proposal did not present any advantage for any algorithm.

4 Conclusion

The evaluation was carried out by averaging the results of 10 independent runs for each training percentage on each dataset. This evaluation approach was chosen because applying 10-fold cross-validation while controlling the training size would significantly reduce the test set, potentially increasing the variance of the results. By using repeated experiments with different random seeds, the chosen evaluation method provides not only greater flexibility in controlling training proportions, but also offers a statistical advantage by producing more independent estimates than traditional cross-validation [5].

Based on the case studies analysed, the proposed approach generally outperforms traditional backpropagation in terms of accuracy. Several cases were identified where the proposal outperformed its classical counterparts by more than 5%, but above all, for all data sets there was always a configuration applying the proposal that reached the best accuracy values. Regarding computational cost, the results are mixed, with some classic configurations sometimes being more efficient than the proposal. It should be noted that for each data set except Glue, there was an architecture that was superior in terms of accuracy percent-

age and also highly efficient in terms of computational cost when comparing the proposed configuration with the classic configurations.

Although our current experiments focus on models trained from scratch, the sector-wise backpropagation paradigm can, in principle, be extended to pre-trained architectures such as BERT or RoBERTa. For instance, transformer blocks could be grouped into few sectors, each augmented with an auxiliary head during training. This design would allow partial fine-tuning with isolated gradients, potentially mitigating overfitting and reducing compute load. We leave a thorough exploration of this integration for future work, as it could open the door to modular fine-tuning techniques for large language models. Moreover, since the comparison was limited to small training percentages of the datasets, further assessment is needed in scenarios with greater data availability, where traditional methods might behave differently.

Acknowledgments. This work was supported by the CONACYT, Paraguay, under Grant PINV01-401.

Disclaimer. During the preparation of this work the authors used generative tools in order to fix misspellings and improve writing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Code Availability. The code for reproducing the experiments presented in this paper is publicly accessible at <https://gitlab.com/pinv01-401/dloptimizer>.

References

1. Agüero-Torales, M.M., López-Herrera, A.G., Vilares, D.: Multidimensional affective analysis for low-resource languages: a use case with Guarani-Spanish code-switching language. *Cogn. Comput.* **15**(4), 1391–1406 (2023)
2. Alain, G., Bengio, Y.: Understanding intermediate layers using linear classifier probes. arXiv preprint [arXiv:1610.01644](https://arxiv.org/abs/1610.01644) (2016)
3. Belilovsky, E., Eickenberg, M., Oyallon, E.: Greedy layerwise learning can scale to ImageNet. In: International Conference on Machine Learning, pp. 583–593. PMLR (2019)
4. Ben Zaken, E., Goldberg, Y., Ravfogel, S.: BitFit: simple parameter-efficient fine-tuning for transformer-based masked language-models. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 1–9. Association for Computational Linguistics, Dublin (2022). <https://doi.org/10.18653/v1/2022.acl-short.1>, <https://aclanthology.org/2022.acl-short.1/>
5. Bengio, Y., Grandvalet, Y.: No unbiased estimator of the variance of k-fold cross-validation. *J. Mach. Learn. Res.* **5**(Sep), 1089–1105 (2004)
6. Chollet, F.: Bidirectional LSTM on IMDB (2020). https://keras.io/examples/nlp/bidirectional_lstm_imdb/, https://keras.io/examples/nlp/bidirectional_lstm_imdb/

7. Clark, J.H., Garrette, D., Turc, I., Wieting, J.: CANINE: pre-training an efficient tokenization-free encoder for language representation. *Trans. Assoc. Comput. Linguist.* **10**, 73–91 (2022). https://doi.org/10.1162/tacl_a_00448, <https://aclanthology.org/2022.tacl-1.5/>
8. Duan, S., Principe, J.C.: Training deep architectures without end-to-end backpropagation: a survey on the provably optimal methods. *IEEE Comput. Intell. Mag.* **17**(4), 39–51 (2022)
9. Gururangan, S., et al.: Don't stop pretraining: adapt language models to domains and tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8342–8360. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.acl-main.740>, <https://aclanthology.org/2020.acl-main.740/>
10. Houshy, N., et al.: Parameter-efficient transfer learning for NLP. In: *Proceedings of the 36th International Conference on Machine Learning. Proceedings of Machine Learning Research*, vol. 97, pp. 2790–2799. PMLR (2019). <https://proceedings.mlr.press/v97/houshy19a.html>
11. Howard, J., Ruder, S.: Universal language model fine-tuning for text classification. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339. Association for Computational Linguistics, Melbourne (2018). <https://doi.org/10.18653/v1/P18-1031>, <https://aclanthology.org/P18-1031/>
12. Iandola, F., Shaw, A., Krishna, R., Keutzer, K.: SqueezeBERT: what can computer vision teach NLP about efficient neural networks? In: *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pp. 124–135. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.sustainlp-1.17>, <https://aclanthology.org/2020.sustainlp-1.17/>
13. Jiao, X., et al.: TinyBERT: distilling BERT for natural language understanding. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4163–4174. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.findings-emnlp.372>, <https://aclanthology.org/2020.findings-emnlp.372/>
14. Maas, A., Daly, R.E., Pham, P.T., Huang, D., Ng, A.Y., Potts, C.: Learning word vectors for sentiment analysis. In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 142–150 (2011)
15. Marivate, V., et al.: Investigating an approach for low resource language dataset creation, curation and classification: Setswana and Sepedi. In: *Proceedings of the first workshop on Resources for African Indigenous Languages*, pp. 15–20. European Language Resources Association (ELRA), Marseille (2020). <https://aclanthology.org/2020.rail-1.3/>
16. Matthew, E., et al: Deep contextualized word representations. In: *Proceedings of NAACL*, vol. 5 (2018)
17. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: *Advances in Neural Information Processing Systems*, vol. 26 (2013)
18. Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., Gao, J.: Deep learning-based text classification: a comprehensive review. *ACM Comput. Surv. (CSUR)* **54**(3), 1–40 (2021)
19. Nandan, A.: Text classification with transformer (2020). https://keras.io/examples/nlp/text_classification_with_transformer/, https://keras.io/examples/nlp/text_classification_with_transformer/

20. Omerick, M., Chollet, F.: Text classification from scratch (2019). https://keras.io/examples/nlp/text_classification_from_scratch/, https://keras.io/examples/nlp/text_classification_from_scratch/
21. Rahamim, A., Uziel, G., Goldbraich, E., Anaby Tavor, A.: Text augmentation using dataset reconstruction for low-resource classification. In: Findings of the Association for Computational Linguistics: ACL 2023, pp. 7389–7402. Association for Computational Linguistics, Toronto (2023). <https://doi.org/10.18653/v1/2023.findings-acl.466>, <https://aclanthology.org/2023.findings-acl.466/>
22. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**(11), 2673–2681 (1997). Introducción formal de redes recursivas bidireccionales (BRNN)
23. Socher, R., et al.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pp. 1631–1642 (2013)
24. Sun, Z., Yu, H., Song, X., Liu, R., Yang, Y., Zhou, D.: MobileBERT: a compact task-agnostic BERT for resource-limited devices. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 2158–2170. Association for Computational Linguistics, Online (2020). <https://doi.org/10.18653/v1/2020.acl-main.195>, <https://aclanthology.org/2020.acl-main.195/>
25. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, vol. 30 (2017)
26. Voorhees, E., Harman, D., Wilkinson, R.: The sixth text retrieval conference (TREC-6). In: The Text REtrieval Conference (TREC), vol. 500, p. 240. ERIC (1998)
27. Waibel, A., Hanazawa, M., Hinton, G., Shikano, K., Lang, K.J.: Phoneme recognition using time-delay neural networks. *IEEE Trans. Acoust. Speech Signal Process.* **37**(3), 328–339 (1989). Primer uso de convoluciones temporales 1D con weight sharing y backpropagation
28. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., Bowman, S.R.: GLUE: A multi-task benchmark and analysis platform for natural language understanding. In: Proceedings of ICLR (2019)
29. Zhang, X., Zhao, J., LeCun, Y.: Character-level convolutional networks for text classification (2015)
30. Zhuang, L., Wayne, L., Ya, S., Jun, Z.: A robustly optimized BERT pre-training approach with post-training. In: Li, S., et al. (eds.) Proceedings of the 20th Chinese National Conference on Computational Linguistics, pp. 1218–1227. Chinese Information Processing Society of China, Huhhot (2021). <https://aclanthology.org/2021.ccl-1.108/>
31. Zulfarnain, M., et al.: Text classification using deep learning models: a comparative review. *Cloud Comput. Data Sci.* 80–96 (2024)



High-Frequency Multiword Units and the Typological Distribution of Multiword Units in Spoken Russian

Natalia V. Bogdanova-Beglarian¹ , Olga V. Blinova^{1,2} , Maria V. Khokhlova¹ ,
Tatiana Y. Sherstinova^{1,2}, and Tatiana I. Popova^{1,2}

¹ Saint Petersburg State University, Saint Petersburg, Russia
{n.bogdanova,o.blinova,m.khokhlova,t.sherstinova,
t.i.popova}@spbu.ru

² HSE University, Saint Petersburg, Russia

Abstract. Multiword units (MWUs) constitute a distinct class of linguistic phenomena located at the crossroads of lexis and syntax. Empirical data on their typology and frequency are essential for solving a wide range of applied problems in natural language processing. This paper presents a corpus-based study of MWUs in Russian everyday speech. Drawing on data from the ORD corpus comprising one million words of transcribed spontaneous discourse, over 8,000 MWU instances were identified and annotated. These MWUs are classified into eight main classes: non-phraseologized collocations, phraseologized collocations, occasional collocations, idiom forms, constructions, precedent texts and their elements, multiword pragmatic markers, and speech formulas. The paper presents a ranked list of the 50 most frequent MWUs in spoken Russian, along with the overall distribution of MWU types. The results indicate that pragmatic markers are the most dominant category (comprising over 30% of all MWUs), followed by non-phraseologized collocations (26%) and speech formulas (21%). The article also discusses the functional combinations of MWUs in spoken interaction and highlights precedent texts as one of the productive sources for MWU formation. The quantitative data obtained in this study contribute to both theoretical models of lexical and grammatical description of Russian everyday speech and practical tasks related to processing and generating spontaneous spoken language.

Keywords: Modern Russian · Everyday Speech · Oral Discourse · Multiword Units · Collocations · Pragmatic Markers · Precedent Texts · Statistical Analysis · Speech Corpus · Corpus Linguistics · Speech Technologies

1 Introduction

Multiword units (MWUs) represent one of the most significant objects of contemporary linguistic research, as they occupy a central position in the study of language both as a system and as a means of communication [1]. They also play a key role in language acquisition and constitute a substantial part of the mental lexicon [2]. The importance of

studying MWUs lies in their special status within cognitive mechanisms responsible for storing and processing linguistic information in the human mind [3]. Recent psycholinguistic studies suggest that MWUs are often processed as holistic units retrieved directly from memory, rather than being decomposed into their individual components [4]. This challenges traditional boundaries between lexis and syntax, showing that the formation of multiword expressions can precede and facilitate the acquisition of individual lexical items. Moreover, MWUs are often culturally marked, reflecting the specific worldview embedded in a given language. Their study offers insight into the deep interconnections between language, cognition, and culture, making MWU research an interdisciplinary field that bridges linguistics (including computational linguistics) and cognitive science. The future prospects of MWU research are closely linked to the development of contemporary technological and methodological approaches in linguistics [5].

Various types of MWUs are actively used in Russian everyday discourse. They have traditionally attracted close attention from linguists, yet many of them still await systematic identification, classification, and detailed description. As noted by researchers, “The presence of a large number of fixed and recurring expressions in the speech practices of speakers of any language is a well-established fact. Phenomena of this kind—referred to as ‘idioms’, ‘fixed phrases’, ‘speech formulas’, ‘clichés’, or ‘templates’—occupy a recognized place in any linguistic description” [6].

In Russian linguistics, this growing interest has led, among other developments, to the creation of specialized databases such as *the Russian Constructicon* [7] and *the Pragmaticon* [8]. These resources are focused on units larger than a single word—constructions and discourse formulas—which occupy the boundary between the lexicon and grammar, effectively challenging the distinction between the two.

The interest in MWUs has also resulted in a number of dedicated studies. The present article continues a larger research project aimed at the systematic classification of MWUs in everyday Russian speech based on the ORD corpus [9, 10]. The ORD corpus was compiled using a methodology of multi-hour audio monitoring of native Russian speakers with diverse social and psychological backgrounds, recorded in a wide variety of communicative contexts. The corpus comprises 1,450 h of audio recordings, speech from 128 informants and over 1,000 interlocutors, and approximately 1 million word tokens in transcribed form (see: [11–14]).

The principles underlying the data collection for the ORD corpus made it possible to obtain speech material that closely approximates natural communication, capturing the richness, spontaneity, “irregularity,” and variability of spoken language. Based on the empirical data of the ORD corpus, a typology of MWUs was developed, as described in [10]. The typology distinguishes eight major classes:

1. Non-phraseologized collocations – semantically compositional fixed word combinations.
2. Phraseologized collocations – semantically non-compositional fixed word combinations.
3. Occasional collocations – modified forms of common collocations (including both phraseologized and free combinations).

4. Idiom forms – frequent combinations of content and function words in everyday speech that have acquired specific meanings and functions (e.g., *ne gorit* “It’s not urgent,” *ne v kayf* “not enjoying it”).
5. Constructions – frequent conventional combinations that include a fixed element (the “anchor”) and a variable slot, whose overall meaning or form is not fully derivable from the semantics of the parts or general grammatical rules [15].
6. Precedent texts and their elements – quotations or fragments of culturally or socially significant texts recognizable in specific contexts.
7. Multiword pragmatic markers – units that do not encode propositional meaning but fulfill various functions related to discourse organization and signaling speaker intention.
8. Speech formulas – multiword expressions typically classified as interjections, expressing the speaker’s emotional reaction or serving as stereotypical responses in dialogue.

This study presents statistical data on the frequency of MWU usage in Russian everyday speech based on a representative volume of transcripts. It identifies the most frequent MWUs, provides quantitative data on the distribution of MWU types, and describes the most typical representatives of each class.

2 Research Material and General Statistics

The statistics presented in this article are based on a sample of 1 million word tokens, representing manually transcribed records of everyday Russian speech from the ORD corpus. The data comprise 500 macroepisodes of everyday spoken communication, covering a wide range of communicative settings (e.g., conversations at home, in the workplace, at educational and medical institutions, in shops, service centers, on the street, etc.). MWUs were identified in 479 macroepisodes (96% of cases).

This analysis was preceded by two earlier studies: (1) an expert annotation of a 300,000-token subcorpus conducted by several annotators, and (2) an automatic annotation of MWUs based on a 700,000-token subcorpus, followed by manual verification. The principles of empirical MWU annotation are detailed in [10], while [16] describes the MWU automatic detection methods and the procedures for expanding the MWU dictionary. All resulting annotated data were subsequently verified manually. In total, 8,055 MWU instances were identified in the speech of 882 individuals—120 primary informants and 762 interlocutors, including both men and women from diverse age and professional groups.

Statistical analysis revealed that MWUs are highly frequent in spoken interaction: their proportion per speech episode ranges from 0% to approximately 3.5%, with an average rate of 1.98% across the sample. The mean length of an MWU is 2.60 orthographic words. A statistically average macroepisode, lasting 15–20 min, contains around 16 MWUs—that is, approximately one unit per minute.

The identified MWUs display considerable heterogeneity. The distribution of the main MWU classes is detailed in Sect. 5. As for formal structure, the most productive morphological patterns observed in the formation of MWUs are as follows:

1. PR S (9.34%; *v printsipe, po idee*) – preposition + noun,
2. APRO APRO (6.58%; *eto samoe, vsyo takoe*) – pronominal adjective + pronominal adjective,
3. PR APRO S (3.36%; *na kakiye shishi, po vashey chasti*) – preposition + pronominal adjective + noun,
4. PART ADVPRO PART (2.94%; *vot tak vot, nu kak zhe*) – particle + pronominal adverb + particle,
5. APRO S (2.89%; *vsyo vremya, takie dela*) – pronominal adjective + noun,
6. SPRO A (2.63%; *nichego strashnogo, sebe dorozhe*) – pronominal noun + adjective,
7. PART APRO PART (2.50%; *vot etot vot, vot tak vot*) – particle + pronominal adjective + particle.

These figures should be interpreted with caution, however, as the original part-of-speech characteristics of components are often blurred in certain MWU classes (e.g., pragmatic markers, speech formulas, phraseologized collocations, and idiom forms).

Precedent texts and their fragments form a separate group within the MWUs. They tend to be longer on average and occur less frequently. Section 7 provides an overview of the most common sources of this MWU type.

The following section presents the most frequent MWUs found in Russian everyday speech.

3 High-Frequency Multiword Units in Spoken Russian

Table 1 presents the top segment of the frequency list of MWUs. The column “ipm” (items per million) indicates the frequency of each unit relative to the total number of word tokens in the dataset (1 million), while the percentage (%) reflects the share of each unit in relation to the total set of identified MWUs.

As shown in Table 1, the three most frequent MWUs are *eto samoe* (“you know”/“um”/“the thing is”), *v printsipe* (“basically”), and *vot tak vot* (“just like that”/“this way”), which together account for 9.60% of all MWU occurrences. These expressions may be considered “core building blocks” of spoken discourse, especially in reflective or impulsive communication—that is, speech not consciously controlled by the speaker and produced at the level of automatized verbal behavior.

Let us now examine the ten most frequent MWUs in greater detail:

1. *Eto samoe* (“you know”/“um”/“the thing is”) is most often used as a hesitation filler, allowing the speaker to gain time for lexical retrieval or formulation of the upcoming utterance. According to the *Dictionary of Pragmatic Markers* [17], this MWU may function as a hesitation pause filler, a self-correction marker, and a discourse boundary marker—serving at all three levels: initial, navigational, and final.
2. *V printsipe* (“basically”) signals generalization or concession. It is used to soften assertions or introduce flexibility into a statement and can express a willingness to compromise. In oral speech, it often functions as an idiom form, although [17] also classifies it as a hesitation and a boundary marker, primarily navigational. Its high degree of polyfunctionality is particularly notable.

Table 1. The 50 most frequent MWUs.

Rank	Multiword units	ipm	%	Rank	Multiword units	ipm	%
1	eto samoe (you know/um)	312	3.87	26	imeyu v vidu (I mean)	35	0.43
2	v printsipe (basically)	290	3.60	27	kak by (sort of/kind of)	34	0.42
3	vot tak vot (just like that/this way)	170	2.11	28	vot eta vot (this one right here [fem.])	33	0.41
4	v obshchem (overall)	167	2.07	29	ty chto (are you kidding?)	33	0.41
5	vsyo ravno (anyway/doesn't matter)	161	2.00	30	v lyubom sluchaye (in any case)	32	0.40
6	kak raz (exactly/just the right moment)	101	1.25	31	vot etot vot (this one right here [masc.])	32	0.40
7	vot eto vot (this one right here)	99	1.23	32	ne znayu (I don't know)	32	0.40
8	i tak dalee (and so on/etc.)	98	1.22	33	nichego strasnogo (it's okay/no big deal)	31	0.38
9	na samom dele (actually)	96	1.19	34	nu i chto (so what?)	31	0.38
10	vsyo vremya (all the time)	91	1.13	35	o Gospodi (oh Lord)	31	0.38
11	tak skazat' (so to speak)	91	1.13	36	vot zdes' vot (right here)	30	0.37
12	nu vot (well then/so)	88	1.09	37	oy Gospodi (oh God)	29	0.36
13	v smysle (I mean)	66	0.82	38	etu samuyu (that one [fem.])	29	0.36
14	nu ladno (alright then/okay)	65	0.81	39	kak ego (what's his name/what's it called)	28	0.35
15	slava Bogu (thank God)	63	0.78	40	na vsyakiy sluchay (just in case)	27	0.34
16	nichego sebe (wow/no way)	62	0.77	41	sovershenno verno (absolutely right)	27	0.34

(continued)

Table 1. (continued)

Rank	Multiword units	ipm	%	Rank	Multiword units	ipm	%
17	etot samyy (you know/that one...)	62	0.77	42	tipa togo chto (something like/kind of like)	27	0.34
18	da ty chto (are you serious?)	48	0.60	43	kak govoritsya (as they say)	26	0.32
19	vot eti vot (these ones)	47	0.58	44	ukh ty (whoa/wow)	25	0.31
20	da ladno (come on!)	47	0.58	45	v kontse kontsov (after all)	24	0.30
21	eti samye (those ones/you know which)	45	0.56	46	vsyo v poryadke (everything's fine)	24	0.30
22	delo v tom chto (the point is that)	43	0.53	47	esli chto (just in case/if anything)	24	0.30
23	Bozhe moy (oh my God)	41	0.51	48	kakaya raznitsa (what's the difference)	24	0.30
24	po idee (theoretically/I suppose)	40	0.50	49	skazhem tak (let's put it this way)	24	0.30
25	po krayney mere (at least)	38	0.47	50	ya ne znayu (I don't know)	24	0.30

3. *Vot tak vot* (“just like that”/“this way”) functions as an affirmative and generalizing phrasal marker, often used to complete or emphasize an action, description, or situation. It is a prototypical structural variant of a deictic pragmatic marker—a descriptive MWU with a pointing function, consisting of three consecutive deictic elements, following the pattern *vot (...) vot*. According to [17], it may also serve as a hesitation marker and, less commonly, as a discourse boundary marker.
4. *V obshchem* (“overall”) is used to summarize, generalize, or shift the topic. Functionally, it acts as a speech structuring device. According to [17], it can function as a hesitation marker, a boundary marker of all three types (initial, navigational, and final), and occasionally as a self-correction marker.
5. *Vsyo ravno* (“anyway”/“doesn’t matter”) expresses indifference, lack of preference, or fatalism. It may be used to close a discussion or to indicate that a choice is of no consequence.
6. *Kak raz* (“exactly”/“just the right moment”) signals precise alignment or congruence with a situation, time, or condition. It adds assertiveness and precision to the speaker’s description.
7. *Vot eto vot* (“this one right here”) is a deictic expression that draws attention to a specific object, action, or event. It is another structural variant of the deictic MWU

vot (...) vot, and similarly functions as a hesitation marker and, more rarely, as a boundary marker. The fact that two structural variants of the *vot (...) vot* model appear among the top ten most frequent MWUs highlights its high productivity in spoken Russian.

8. *I tak dalee* (“and so on”/“etc.”) is a generalizing marker used to conclude a list or indicate a logical continuation. It simplifies speech and, as a pragmatic marker, serves primarily as a substitutive boundary device.
9. *Na samom dele* (“actually”/“in fact”) emphasizes the truthfulness or relevance of a statement. It is often used to highlight the importance or unexpected nature of a fact.
10. *Vsyo vremya* (“all the time”) denotes the duration or regular recurrence of an action. It emphasizes frequency or repetition.

In summary, the most frequent MWUs in Russian everyday speech are predominantly pragmatic markers characterized by hesitation, deixis, and discourse structuring functions. Their high frequency highlights the importance of not only propositional but also metadiscursive organization in spoken interaction—the speaker’s ability to navigate communicative contexts, manage the interlocutor’s attention, and structure their own speech.

4 Invariant Forms of Multiword Units

As Table 1 clearly shows, some MWUs—even frequent ones—can be interpreted as variants of a more general unit, or invariant. For example, *vot tak vot*, *vot eto vot*, *vot eti vot*, *vot eta vot*, *vot etot vot*, *vot zdes’ vot*, *vot takoj vot*, etc., evidently belong to the *vot DEIX vot* group, in which two occurrences of the particle *vot* frame a deictic element (such as “this,” “here,” “such,” etc.). Invariants typically represent structural and semantic templates, within which certain components—most commonly deictic or evaluative elements—vary.

Invariants were identified for 54% of the total number of MWU forms (3,466 tokens in total), resulting in a list of 1,815 distinct invariant types. The high proportion of such invariant-based expressions indicates a significant degree of regularity in the organization of spontaneous speech, despite its apparent disorder and variability. However, identifying invariant forms is not always straightforward [18].

Table 2 presents the 10 most frequent MWU invariants in Russian, including both their overall frequency in items per million (ipm) relative to all word tokens in the analyzed subcorpus, and their proportion (%) relative to all identified MWUs.

Individual MWUs may either fully coincide with their invariant form (e.g., *nu vot* or *slava Bogu*), represent a structural variation of it (as in the example of *vot DEIX vot*), or function as occasionalisms—unique, rarely occurring expressions.

To study rare MWUs, we can refer to the hapax list, which includes all units that occurred only once in the dataset. In our sample, hapax MWUs account for 2,320 types, representing 28.80% of all MWUs in the frequency list.

Thus, the analysis of invariant forms makes it possible to identify productive patterns underlying the formation of stable expressions in spoken language. The high percentage of hapax units—unique or occasional combinations—reflects the creative potential of Russian everyday speech.

Table 2. Most frequent invariants of MWUs in Russian everyday speech.

Rank	Invariants Multiword units	Share of tokens (ipm)	Share of MWU (%)
1	*eto *samoe (this... thing)	292	5.60
2	vot DEIX vot (here it is/you know/this one)	183	3.51
3	v printsipe (basically)	170	3.26
4	nu vot (well then/so)	84	1.61
5	v obshchem (i tselom) (in general (and overall))	74	1.42
6	vsyo ravno (doesn't matter/anyway)	66	1.27
7	nu ladno (alright then)	61	1.17
8	i tak daleye (and so on)	55	1.06
9	*imet' v vidu (to mean/to have in mind)	42	0.81
10	tak skazat' (so to speak)	40	0.77
11	slava Bogu (thank God)	36	0.69
12	(*...) ty *chto (what are you.../are you serious?)	35	0.67
13	(*) Gospodi (oh Lord/dear God)	34	0.65
14	X sebe (wow/damn/oh my)	34	0.65
15	kak by (like)	34	0.65

5 Distribution of MWU Classes in Everyday Russian

This section discusses the distribution of individual MWU classes in Russian everyday spoken speech. As shown in the frequency list of MWUs presented in Sect. 3, pragmatic markers occupy the leading position among all MWU types in terms of frequency. The statistical data confirm this observation: pragmatic markers account for 2,487 instances, or 30.88% of all identified MWUs. Below, we take a closer look at the distribution of the main MWU classes (Table 3).

1. Multiword pragmatic markers (PMs) top the frequency list of MWUs in everyday Russian spoken communication. These markers play a crucial role in structuring discourse, helping speakers manage the flow of conversation and cope with communication difficulties or disfluencies. As shown in Sect. 3, most of the highest-frequency MWUs fall into this category, including: (1) *eto samoe* (“you know”) (312 ipm; 3.87%)¹, (2) *v printsipe* (“basically”) (290; 3.60), (3) *vot tak vot* (“just like that”) (170; 2.11), (4) *v obshchem* (“overall”) (166; 2.06), and (5) *vot eto vot* (“this one right here”) (99; 1.23), among others.

¹ Throughout this section, frequencies are reported in the following format: (occurrences per million words [ipm], representing the share of a given MWU type in the total word count of the corpus; percentage [%], representing its share among all identified MWUs).

Table 3. Distribution of MWU classes in spoken Russian.

Rank	MWU Classes	Share of tokens (ipm)	Share of MWU (%)
1	Pragmatic markers (PM)	2487	30.88
2	Non-phraseologized collocations (NK)	2119	26.31
3	Speech formulas (RF)	1673	20.77
4	Phraseologized collocations (FK)	800	9.93
5	Idiom forms (ID)	447	5.55
6	Constructions (KS)	407	5.05
7	Precedent texts (PT)	101	1.25
8	Occasional collocations (OK)	19	0.24

- Non-phraseologized collocations (NK) rank second in frequency, collectively accounting for more than one-quarter of all MWUs. The most frequent among them include: (1) *kak raz* (“exactly/right on time”) (101; 1.25), which denotes precision or timeliness; (2) *na samom dele* (“actually/in fact”) (96; 1.19), used to emphasize the truthfulness or factuality of a statement; (3) *vsyo vremya* (“all the time”) (91; 1.13), indicating repeated or continuous action; (4) *vsyo ravno* (“anyway/regardless”) (90; 1.12), which expresses indifference or inevitability; and (5) *po krayney mere* (“at least”) (38; 0.47), introducing a minimum estimate or condition.
- Speech formulas (RF) are the third most frequent class, comprising approximately one-fifth of all MWUs. These units are characterized by their use in expressing emotions, reactions, or standard conversational routines in everyday speech. The most typical examples include: (1) *nu vot* (“well then”) (67; 0.83), often used as an introductory or closing phrase to mark transitions or conclusions—it can also function as a navigational pragmatic marker; (2) *slava Bogu* (“thank God”) (67; 0.83), conveying relief or gratitude for a positive outcome; (3) *nichego sebe* (“no way”/“wow”) (62; 0.77), an expression of surprise or admiration; (4) *da ty chto* (“are you serious?”/“really?”) (48; 0.60), an emotional and expressive marker used to signal surprise, disbelief, admiration, indignation, or disagreement; and (5) *da ladno* (“come on”/“seriously?”) (46; 0.57), whose interpretation depends on context—may indicate disbelief, encouragement, resignation, or amused acceptance.
- Phraseologized collocations (FK) account for approximately 10% of all MWUs, although their absolute and relative frequencies remain low. The most common examples include: (1) *ne day Bog* (“God forbid”) (9; 0.11), expressing fear or strong undesirability; (2) *day Bog* (“God willing”) (7; 0.09), used to express hope or a wish for good fortune; (3) *s uma soyti* (“to go crazy”) (5; 0.06), conveying extreme surprise or shock; (4) *chort-te chto* (“complete mess”) (5; 0.06), a reaction to something absurd or chaotic; and (5) *vyshe kryshi* (“more than enough”) (3; 0.62), denoting abundance or excess.
- Idiom forms (ID) represent 5.55% of all MWUs. These include colloquial expressions such as *bez problem* (“no problem”) (5; 0.06), *v nature* (“seriously”/“really”) (5; 0.06), and *po idee* (“theoretically”/“supposedly”) (3; 0.04).

6. Constructions (KS) are close in frequency to idiom forms, accounting for 5.05% of all MWUs. Examples include *Bog s nim* (“let it go”/“so be it”) (8; 0.10), an expression of acceptance or resignation, and *vot takie dela* (“that’s the way it is”) (6; 0.05), often used to close a topic with a note of resignation, irony, or quiet reflection.
7. Precedent texts (PT) are even rarer, comprising only 1.25% of the total. Just six expressions occurred twice in the entire corpus, including: (1) *vremya pokazhet* (“time will tell”) – don’t rush to conclusions; (2) *vremya sobirat’ kamni* (“time to gather stones”) reaping the consequences of one’s actions; (3) *eshchyо ne vecher* (“the night is still young”) meaning things might still change; (4) *na sebya lyubimuyu* (“on myself, my dear self”) meaning spending time/money/care on oneself, often humorously; (5) *pozдно pit’ Borzhomi* (“too late to drink Borjomi”) – it’s too late to fix something; and (6) *khorosho sidim* (“we’re having a good time”) expresses the speaker’s satisfaction with the current moment and is typically used in informal or celebratory settings (e.g., during a friendly gathering or a shared meal).
8. Finally, occasional (non-conventional) collocations (OK) are the rarest category: only 19 such units were identified in the 1-million-word corpus, accounting for just 0.24%. Examples include highly creative or humorous phrases like *kushat’ batareyku kak traktor* (“to eat a battery like a tractor”).

Overall, the distribution of MWU classes in everyday spoken Russian reveals a strong dominance of pragmatic markers, non-phraseologized collocations, and speech formulas. These categories play a central role in organizing spontaneous communication by structuring discourse, expressing speaker attitudes, and ensuring fluency. At the same time, the presence of less frequent but semantically rich types—such as idioms, phraseologisms, constructions, and precedent text elements—highlights the expressive diversity and dynamic nature of colloquial Russian.

6 Multiword Units with Dual Class Annotations

From the beginning of the study, the development of the typology of multiword units (MWUs) proposed in [10] has presented significant challenges. These units are not only often polysemous and multifunctional within a single category, but they can also be interpreted in more than one way within the typological framework itself.

For instance, the MWUs *bez voprosov* (“no questions”), *bez problem* (“no problem”), and *bez bazara* (“no worries”) can be reasonably classified either as idiom forms (ID) (*bez* + N2) or as constructions (KS) (*bez* + X) [12]. The precedent text (PT) *ne syp’ mne sol’ na ranu* (“don’t rub salt in my wound”) may be interpreted in contemporary usage as a phraseologized collocation (PC), while the occasional collocation (OC) *kto riskuyet, tot p’yot shampanskoye* (“those who risk, drink champagne”) is a modification of the precedent text *kto ne riskuyet, tot ne p’yot shampanskogo* and can be seen as its variation. Analyzing the material from this perspective has yielded some noteworthy observations.

In total, 320 MWU instances (3.97%) in the annotated corpus were found to belong to more than one category. More specifically, the following picture emerged:

1. The most frequent overlap was found between speech formulas (RF) and non-phraseologized collocations (NK), accounting for 20.63% of all cases of MWUs

with dual class annotations. These MWUs operate both as conversational clichés commonly used in everyday interactions and as fixed lexical combinations aligned with standard language usage.

2. The combination of pragmatic markers (PM) and non-phraseologized collocations (NK) occurred in 8.75% of cases. These units serve both as discourse-structuring units and as conventionalized lexical combinations.
3. The overlap between phraseologized collocations (FK) and occasional collocations (OK) accounted for 10.63%. These MWUs combine the fixed nature of idiomatic phrases with the contextual uniqueness typical of occasional usages.
4. The combination of non-phraseologized (NK) and phraseologized (FK) collocations accounted for 7.50%. These MWUs illustrate a shift from structurally stable word combinations to idiomatic expressions.
5. The overlap between constructions (KS) and speech formulas (RF) occurred in 10.31% of cases. These units merge the grammatical regularity of constructions with the conventionality of common conversational routines.
6. The combination of pragmatic markers (PM) and speech formulas (RF) occurred in 7.19% of cases. Such MWUs fulfill both discourse-organizing and conventional expressive functions.
7. The pairing of idiom forms (ID) and speech formulas (RF) accounted for 5.63%. These MWUs combine figurative meaning with formulaic usage typical of conversational exchanges.
8. Finally, the combination of non-phraseologized collocations (NK) and idiom forms (ID) made up 5.63%, reflecting stable expressions that may include elements of figurative meaning.

It is thus evident that the most frequent overlap occurs between speech formulas (RF) and non-phraseologized collocations (NK), accounting for 20.63% of all dual-function cases—highlighting their prevalence in spoken communication. The affinity between these two MWU types can be explained by the fact that RFs are both conventionalized speech clichés and word combinations with statistically high co-occurrence. As a result, many RFs are actually formed from non-phraseologized collocations. For example, *Kak dela?* (“How are you?”) is a speech formula (specifically, a greeting formula), but it is also a collocation. Such MWUs are frequently used in everyday dialogues, where predictability and automatism are important.

High proportions are also observed in the overlap between phraseologized collocations (FK) and occasional collocations (OK) (10.63%), and between constructions (KS) and speech formulas (RF) (10.31%). These figures point to the functional flexibility of these categories. The latter overlap, for instance, may be attributed to the fact that many RFs are based on constructions, since the reproducible structure of the latter facilitates predictability and ease of comprehension.

In a similar way, the motivation behind other observed combinations of MWU classifications can also be explained. However, the very existence of such overlaps confirms that many MWUs have a complex nature and simultaneously perform multiple communicative functions.

7 Prototypes and Sources of Selected Multiword Units

The analysis also identified prototypes of MWUs—original, “primary” forms of multiword units that emerged or became entrenched in the language through specific cultural sources familiar to native speakers. A prototype defines the form, intonation, and meaning of a given MWU, and may later be modified or used in new contexts. Identifying prototypes is especially relevant for MWUs that function as elements of precedent texts. The following are among the main cultural sources and their corresponding prototypes:

1. Quotations from works of Russian classical literature. For example, the MWU “*ya ne khochu uchit'sya, khochu zhenit'sya*” (“I don’t want to study, I want to get married”) corresponds to a line from Denis Fonvizin’s play *The Minor*; cf. also “*staro kak mir*” (“as old as the world”) from Leo Tolstoy’s *Anna Karenina*.
2. Biblical quotations, such as “*vsyo taynoye stanovitsya yavnym*” (“everything hidden shall be revealed”).
3. Quotes from popular films, TV series, and animated cartoons—for example: “*na eto ya poytit' ne mogu*” (“I can’t go along with this”), “*tsigel-tsigel, ay-lyu-lyu*” (from the Soviet comedy *The Diamond Arm*), “*tvoy tuflya*” (“your shoe,” from *Kidnapping, Caucasian Style*), “*zhizn' moya zhestyanka*” (“my life is a tin can,” from the animated film *The Flying Ship*).
4. Proverbs (paremias), including traditional sayings such as “*kashu maslom ne isportish'*” (“you can’t spoil porridge with butter”), modern variants like “*kto riskuyet, tot pyot shampanskoye*” (“those who take risks drink champagne”), and proverbial expressions rooted in Roman classical literature—e.g., “*vsyo svoyo noshu s soboy*” (“I carry all my things with me,”—“*Omnia mea mecum porto*” by Bias of Priene and Cicero), among others.
5. Works of popular musical culture—both Soviet/Russian and Western—e.g., “*nas utro vstrechaet prokhladoy*” (“the morning greets us with a chill,” from *The Song of the Counterplan*) and “*We are the champions*” (from the song by Queen).
6. Political slogans—for instance, Soviet-era slogans such as “*gotov k trudu i oborone*” (“ready for labor and defense”) and “*dayosh proletariat*” (“forward with the proletariat”).
7. Jokes and anecdotes—e.g., the well-known series about the character *Vovochka*, including the phrase “*vsyo, Vovochka, ne snosit' tebe golovy*” (“that’s it, Vovochka, you’re doomed”).
8. Advertising—for example: “*detochka, ty lopnesh'*” (“sweetie, you’ll burst”) or “*zheludok u kotyonka men'she naperstka*” (“a kitten’s stomach is smaller than a thimble”).

Thus, MWUs functioning in everyday Russian speech are linked to a wide range of sources—literature, film, advertising—and serve as indicators of the cultural influence on language.

8 Conclusion

The study examined key patterns in the usage and functions of multiword units (MWUs) in everyday spoken Russian. Based on a one-million-word sample from the ORD corpus, it identified the most frequent MWUs, developed a typology of their forms and functions,

uncovered invariant structures, documented cases of overlapping classifications, and analyzed the cultural sources of emerging MWUs.

The most frequent MWU classes—pragmatic markers, non-phraseologized collocations, and speech formulas—play a key role in structuring spontaneous speech, managing dialogue, and expressing emotions and communicative intentions. At the same time, idioms, constructions, elements of precedent texts, and occasional formations illustrate the expressive richness and creative potential of spoken language.

The findings have high applied value. They can be used in the development of speech recognition and synthesis systems, as well as in the creation of more natural and human-like dialogue systems, including virtual assistants and voice interfaces. The typology and frequency data of MWUs are essential for improving language models, particularly those aimed at processing everyday spoken language, where a high share of “non-standard” units requires special treatment. Empirical study of MWUs brings us closer to building more accurate and adaptive technologies for speech-based interaction and opens possibilities for integrating linguistic insights into computational language processing.

Looking ahead, one of the most promising directions is the automatic extraction of MWUs from corpora using neural networks and deep learning. Such methods are critically important for advancing machine translation technologies, as the inability to automatically recognize MWUs often leads to translation errors and reduces the efficiency of natural language processing systems. Future work should also include the development of multilingual corpora annotated for MWUs, the creation of universal algorithms for MWU identification across languages, and the integration of MWU research into automated systems for assessing text complexity. The interdisciplinary nature of this field ensures its ongoing development and underscores its practical significance for addressing both fundamental issues in language theory and applied challenges in computational linguistics.

Acknowledgments. This research has been carried out thanks to the financial support of Russian Science Foundation (project No. 22-18-00189 “Structure and Functionality of Stable Multiword Units in Russian Everyday Speech”).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Constant, M., et al.: Multiword expression processing: a survey. *Comput. Linguist.* **43**(4), 837–892 (2017). https://doi.org/10.1162/COLI_a_00302
2. Gilquin, G.: The processing of multiword units by learners of English: Evidence from pause placement in writing process data. *Languages* **9**, 51 (2024). <https://doi.org/10.3390/languages9020051>
3. Wray, A.: *Formulaic Language and the Lexicon*. Cambridge University Press, Cambridge (2002)
4. Siyanova-Chanturia, A., Martinez, R.: The idiom principle revisited. *Appl. Linguist.* **36**(5), 549–569 (2015)

5. Columbus, G.: In support of multiword unit classifications: Corpus and human rating data validate phraseological classifications of three different multiword unit types. *Yearb. Phraseol.* 4(1), 23–44 (2013). <https://doi.org/10.1515/phras-2013-0003>
6. Gasparov, B.M.: *Yazyk, pamyat', obraz. Lingvistika yazykovogo sushchestvovaniya* [Language, Memory, Image: The Linguistics of Linguistic Existence]. Novoe literaturnoe obozrenie, Moscow (1996)
7. Bast, R., et al.: *The Russian Construction. An Electronic Database of the Russian Grammatical Constructions* (2021). <https://constructicon.github.io/russian/>
8. Yaskevich, A., et al.: *The Russian Pragmaticon. An Electronic Database of the Russian Pragmatic Constructions* [Electronic resource]. <https://pragmaticon.ruscorpora.ru/>
9. Bogdanova-Beglarian, N., Blinova, O., Khokhlova, M., Sherstinova, T.: Towards the description of multiword units in Russian everyday speech: State-of-the-art and the methodology of further research. In: Bolgov, R., Mukhamediev, R., Pereira, R., Mityagin, S. (eds.) *Digital Geography. IMS 2022*. Springer, Cham (2024). https://doi.org/10.1007/978-3-031-50609-3_10
10. Bogdanova-Beglarian, N., Blinova, O., Khokhlova, M., Sherstinova, T., Popova, T.: Multiword units in Russian everyday speech: empirical classification and corpus-based studies. In: Karpov, A., Delić, V. (eds.) *Speech and Computer: 26th International Conference, SPECOM 2024, Belgrade, Serbia, 25–28 November 2024, Part I*, pp. 187–200. Springer, Cham (2024)
11. Sherstinova, T.: The structure of the ORD speech corpus of Russian everyday communication. In: Matoušek, V., Mautner, P. (eds.) *TSD 2009. LNAI (LNAI)*, vol. 5729, pp. 258–265. Springer, Heidelberg (2009)
12. Bogdanova-Beglarian, N.V. (ed.): *Russkiy yazyk povsednevnogo obshcheniya: osobennosti funktsionirovaniya v raznykh sotsial'nykh gruppakh* [Russian Everyday Language: Patterns of Use in Different Social Groups]. Collective monograph. LAIKA, St. Petersburg (2016)
13. Bogdanova-Beglarian, N.V., Blinova, O.V., Martynenko, G.Ya., Sherstinova, T.Yu.: *Korpus russkogo yazyka povsednevnogo obshcheniya "Odin rechevoy den'": tekushchee sostoyanie i perspektivy* [The ORD Corpus of Russian Everyday Speech: Current State and Prospects]. In: *Trudy Instituta russkogo yazyka im. V. V. Vinogradova*, vol. 21, pp. 101–110. Moscow (2019)
14. Bogdanova-Beglarian, N., Sherstinova, T., Blinova, O., Baeva, E., Martynenko, G., Ryko, A.: Sociolinguistic extension of the ORD corpus of Russian everyday speech. In: *SPECOM 2016. LNAI (LNAI)*, vol. 9811, pp. 659–666. Springer, Cham (2016)
15. Hilpert, M.: *Construction Grammar and Its Application to English*. Edinburgh University Press, Edinburgh (2014)
16. Sherstinova, T., Popova, T.: Multiword units in Russian spontaneous spoken language: methods for vocabulary expansion and statistical analysis. In: *LiLAC-24 (LiLaC – Literature, Language, Computing: Russian Contribution)*. Springer (in print)
17. Bogdanova-Beglarian, N.V. (ed.): *Pragmaticheskie markery russkoy povsednevnoy rechi: Slovar'-monografiya* [Pragmatic Markers in Russian Everyday Speech: Dictionary-Monograph]. Nestor-Istoriya, St. Petersburg (2021)
18. Popova, T.I., Igolkina, A.E.: *Khochesh' ne khochesh', a nado: problema opredeleniya invariantnykh struktur ustoychivyykh neodnoslovnnykh edinits* [Whether You Want It or Not: On the Problem of Defining Invariant Structures of Multiword Units]. In: *IX Mezhdunarodnyi nauchnyi simpozium "Russkaya grammatika: poliparadigmálnost' kak metodologicheskii printsip sovremennykh nauchnykh issledovaniy"*, Irkutsk (2025, in print)



Estimation of the Genre Composition of the English Subcorpus of the Google Books Ngram

Vladimir Bochkarev^(✉) , Andrey A. Achkeev , and Anna Shevlyakova 

Kazan Federal University, Kazan, Russia
Vladimir.Bochkarev@kpfu.ru

Abstract. The Google Book Ngram corpus is captivating due to its incredible size and availability. It has been widely used in studies of culture, social psychology, language evolution and others. However, as some researchers state, it suffers a number of limitations. Apparently, the most serious limitation of the corpus is its genre imbalance and lack of information on the genre composition of the books included in it. In this paper, we developed an algorithm for estimating the genre composition of the Google Books Ngram corpus. To estimate the percentage of texts of different genres, we used data on relative frequencies of a large range of words. Both linear models and multilayer feedforward neural networks were tested as predictors. To train the predictors, we used random subsamples of texts from the COHA corpus which are marked up by genres. We obtained estimates by using linear predictors and neural network predictors which showed different effectiveness. To assess the achieved accuracy, a cross-validation was performed. The analysis showed that the standard deviation of the neural network estimates obtained from annual data is no worse than 2–2.2%. The constructed estimates of the genre composition of Google Books Ngram also response to major historical events. It should also be noted that the genre composition has changed significantly since 2008. The obtained results provide a vision of the Google Book Ngram corpus genre composition and offer a possible framework for improvements to future works based on the Google Book Ngram data.

Keywords: Google Books Ngram · Genre Composition · Neural Networks · Text Corpora

1 Introduction

Development of new technologies triggered creation of large text corpora in different languages. One of such text databases is Google Books. It is a huge collection of digitized book texts which allows searching for usage of words and word combinations. Based on the Google Books project, an open diachronic corpus Google Books Ngram [1] (hereinafter GBN) was compiled, which contains data on frequencies of words and word combinations in 8 languages (American and British English, Russian, Spanish, French, German, Chinese, and Hebrew) [2]. To date, 3 versions of GBN have been created.

The first GBN version was published in 2009 and included over 5 million books in English, largely provided by large university libraries [3]. The second GBN version appeared in 2012 and contained 8 million books. The content of the books of the first and second versions was split into case-sensitive n-grams. N-gram is a part of text including different number of neighbouring words. Thus 1-gram consists of one word, 2-gram consists of two words etc. Starting with the second version of the corpus, syntactic 2-grams also appeared. Syntactic 2-grams are not usually two neighbouring words but they must be in a syntactic relationship with each other. The third version of GBN was published in 2020 and included 16.6 million books in English containing a total of 2 trillion words.

Incredible size and availability of the GBN corpus made it an interesting tool for various types of research in the fields of linguistics, psycholinguistics, psychology, culturology and other sciences [3, 4].

However, despite a large number of investigations performed on the GBN material, there are works that are critical of the corpus, primarily pointing out its genre imbalance [5, 6]. The authors of these works point out that GBN does not contain an unbiased sampling of publications. Thus, the composition of GBN is criticized, for example, in [5]. The article emphasizes that the corpus suffers from a number of limitations. One of the distinctive problematic features they consider is the inclusion of a large number of scientific publications throughout the 1900s. The result is that the corpus is overflowed with academic vocabulary that is rarely used in everyday speech. The only GBN subcorpus that was not heavily affected by professional texts was the English Fiction dataset (2012). The authors criticize the library-like nature of GBN and call for a proper account for the biases of the unfiltered corpus [5, 6]. The corpus was also criticized in [7] for lacking metadata and data truncation and it is suggested not making general conclusions about the evolution of language or culture based on GBN but to make conclusions noting “as it is represented in Google Ngram data”. The imbalance of the corpus is also discussed in [8].

It is obvious that if there was information about the genre composition of the GBN corpus for each year, this would greatly facilitate researchers’ work. However, the creators of the corpus do not provide such information. In [9], estimates of the genre composition of the Russian subcorpus of GBN were proposed. However, the algorithm proposed in [9] is based on the frequency analysis of only 105 words, which reduces the reliability of the results obtained in the work.

Automatic genre identification is one of the acute problems of computational linguistics and related disciplines. Many investigations have been already conducted in this field. A detailed overview of different methods and approaches to automatic genre analysis is presented in [10]. The previous methods use support vector machines (SVMs) [11], discriminant analysis [12], and Naïve Bayes algorithm [13] and others. Developments in NLP allowed researches conducting experiments using word embeddings and neural networks for genre identification [14–16]. Recently, Large Language Models have revolutionized NLP and showed significant improvements in genre classification [17–19].

As for GBN, there is no access to its texts and the above-mentioned approaches cannot be applied. If the GBN texts were available, it would be possible to use existing

algorithms to identify their genres and calculate the percentage of texts of each genre. Instead, it is necessary to directly determine the percentage of texts of different genres using the information available in the public domain. Therefore, in this paper, we propose an algorithm for estimating the genre composition based on the analysis of frequency statistics for a wide range of vocabulary. To train the model, we use The Corpus of Historical American English (COHA) which provides information about the genre of each of the texts included in the corpus.

2 Data and Method

2.1 Corpus of Historical American English

To train the model, we used the Corpus of Historical American English (COHA) [20, 21], created by an American researcher Mark Davis. COHA is currently the largest well marked diachronic corpus of the English language. It contains 400 million words in 100 thousand texts which date from the 1800s to the 2000s. Compared to the GBN, COHA is a small corpus, however, it has other undeniable advantages. It is filtered and balanced by genre from decade to decade including texts from fiction, popular magazines, newspapers and non-fiction books. The texts are marked by genres in accordance with the classification of the Library of Congress of the United States. COHA is carefully lemmatized and tagged for part-of-speech. Being not just a text archive but a carefully structured and balanced corpus, it allows for a wider range of studies of changes in lexis, morphology, syntax, semantics and American society than other historical corpora of American English or text archives.

COHA texts are marked by four genres, which are *Fiction*, *Magazine*, *News* and *Non-Fiction*. Fiction texts are presented by scanned fiction books, movie and play scripts from COCA, Project Gutenberg, and Making of America. Magazine data include texts from magazines balanced across ten magazines [21]. *News* genre is presented by texts from newspapers. Non-fiction texts include non-fiction scanned books and texts from Project Gutenberg, COCA and www.archive.org.

M. Davis [21] points out that “starting in the 1860s, we have very good genre balance from one decade to the next”. For earlier years, the corpus lacks news texts. Therefore, to perform our work, we used the corpus texts for fifteen decades, that is, for the years 1860–2009.

2.2 English Subcorpora in Google Books Ngram

The Google Books Ngram corpus includes four English subcorpora. First, there are the American and British English subcorpora. Second, there is the Common English subcorpus which includes all texts in English. Finally, there is a separate English Fiction corpus. Despite its name, the English Fiction subcorpus apparently includes more than just fiction books, as it contains a large number of specialized scientific, technical, and medical terms. Table 1 shows the size of the English subcorpora in the 2nd and 3rd versions of Google Books Ngram. All results presented below, unless otherwise stated, refer to the 3rd version of the corpus.

Table 1. The size of the English subcorpora in Google Books Ngram.

Subcorpus	The size of the 3 rd version, bln of words	The size of the 2 nd version, bln of words
Common	1997.5	468.5
American	1167.2	356.0
British	337.0	130.2
Fiction	159.0	64.5

A serious limitation of the corpus is that there is no access to its texts. As it was mentioned above, it contains only statistics on word and n-gram usage. Also, there is no information about the genre composition of GBN. The only known thing is that the corpus includes a significant percentage of book ever published, therefore, it contains books of all genres. As the corpus uses book texts, it does not contain news texts (unlike COHA). However, there are books devoted to social and political acute issues which lexicon is close to news texts. Using search tools shows that GBN also includes magazine texts.

2.3 Training Samples

The model is constructed as follows: a vector of relative word frequencies is fed to the input, and a vector of percentage of texts in each genre is obtained at the output. Thus, the first step is to decide on the list of words. Firstly, the list included words consisting of letters of the English alphabet with the possible exception of one apostrophe. Abbreviations with one dot at the end were also allowed (Mr., fig., etc.). Secondly, we selected words that occurred in the COHA corpus in at least 12 decades out of 15. This was done to obtain a list equally representative of each decade for the period under study. The list obtained in this way included 60,910 words. Of this number, 275 were abbreviations with a dot. Experiments were conducted with both the full list and a truncated list without abbreviations (60,635 words).

It is obvious that frequencies of different words are informative to different degrees. Many words, such as function words, are found in the texts of all four types. Therefore, we also prepared truncated lists of the most informative words. For this purpose, the value of mutual information between the presence/absence of a word in a text and the belonging of this text to a particular genre was calculated for each word in the large list. For each word, a 2×4 probability matrix of presence or absence of this word in texts of each of the four genres was calculated. Then, the mutual information was also calculated. 10% of the words with the highest values of mutual information were selected. In this way, two lists were obtained: 6,091 words (including abbreviations with a dot) and 6,065 words (excluding abbreviations with a dot).

Random subsamples of texts from the COHA corpus were used to train the models (similar to the bootstrapping procedure [22]). The model needs to work correctly with a wide range of percentage of texts of different genres, so the sample must contain examples in which these percentage vary within wide limits. The average size of text collections was set at 7 million words. First, the target percentages of texts of different genres were set for each example. To do this, random divisions of a unit segment into

4 parts were generated. After that, a random sample of texts of each type was selected to obtain the specified size. In this case, repeated selection of texts was allowed. Based on the corpus data, 1000 such random sets of texts were generated for each of the 15 decades. This was done so that the model could be trained to work with data for the entire time interval of 1860–2009. A total of 15,000 random samples of texts were obtained in this way. For each of them, the relative frequencies for the given list of words and the percentage of texts of each of the four genres were calculated.

2.4 Cross-Validation Procedure

To control the reliability of the obtained results, a cross-validation procedure was used. We independently trained 15 models on different training subsets and compared the obtained results. The training subsets was divided into 6 groups, of which 4 groups were used each time to train the model, and 2 groups for testing it. There are 15 different ways to select 4 groups out of 6. Thus, we obtained 15 ways of division into a training and testing sets in a two-to-one ratio. Having trained a model independently on each of these subsets, for each example we obtained 6 models (out of 15) for which this example was in the test sample and was not used in the training process. This makes it possible to calculate the standard deviation of the obtained estimates. Moreover, one can additionally increase the accuracy for neural network models by averaging the outputs of those independently trained models for which the example was in the test sample.

2.5 Linear Model

Linear and neural network estimates of the genre composition were tested. In the linear model, the vector of estimates $\hat{p}^{(g)}$ percentage of the genre g is sought in the form (1):

$$\hat{p}^{(g)} = F^H x^{(g)} \quad (1)$$

Here F is an $n \times M$ matrix of relative frequencies of the selected words in each example, n is the number of words in the list, and M is the number of examples. $x^{(g)}$ denotes the n -dimensional vector of model coefficients for genre g , $\hat{p}^{(g)}$ is the M -dimensional vector of estimates of the percentage of text for genre g in M examples. As the model has a high dimensionality, we use L2 regularization. Thus, the model coefficients are found by minimizing the following function:

$$\|p^{(g)} - F^H x^{(g)}\|^2 + \lambda \|x^{(g)}\|^2 \quad (2)$$

From here we obtain the formula for the model coefficients:

$$x^{(g)} = \{F^H F + \lambda I\}^{-1} F^H p^{(g)} \quad (3)$$

2.6 Neural Network Model

In addition to linear classifiers, we also used direct propagation neural networks (multi-layer perceptron). The network input is a vector of word frequencies normalized to a unit sum (with dimensions from 6,065 to 60,910, depending on the selected list of words), and estimates of the percentage of texts of each of the 4 genres are taken from 4 outputs. The network architecture was chosen based on the results of preliminary experiments. For all classifiers, a network with 4 internal layers of 64 nodes with the ReLU activation function and the output layer of 4 neurons with softmax activation were used to ensure the total percentage of all genres equal to 100% [23]. Training was carried out according to the criterion of minimum mean square error.

The training was performed by the Adam algorithm with a learning rate of $5 \cdot 10^{-4}$, the L2 regularization weight was from $8.4 \cdot 10^{-8}$ to $1 \cdot 10^{-6}$ depending on the length of the word list. The stopping criterion was 30 steps without an absolute change in the mean square error by $1 \cdot 10^{-6}$. Trying to increase the number of hidden layers and the number of neurons in the layer did not improve the quality of the model. Stochastic gradient descent was also tested as a training algorithm, and the dropout method [23] was used as a regularization method. However, all these algorithms provided slightly worse results. To build and train the described neural network, the TensorFlow and Keras machine learning libraries were used [24, 25].

3 Results

3.1 Results for COHA

All linear models showed high accuracy on the test set. Standard deviation of error for different models and genres is within the range 2.1–2.39%. At the same time, linear models independently trained on different subsets of examples show high degree of consistency with each other. The correlation coefficients between the estimates for different models on the intersecting part of the test samples are within the range from 0.9991 to 0.9996. Table 2 shows the root mean square difference in estimates of the percentage of texts of different genres obtained by different models (selected from 15 independently trained on different subsets of training examples). The table shows the results for 4 groups of models that differ in the number of words in the list (see Sect. 2.3).

Table 2. Root mean square differences of estimates on the intersecting part of test samples for independently trained models.

Model (number of words)	Root mean square difference, %
60,910	0.54
60,635	0.53
6,091	0.76
6,065	0.78

Therefore, we further synthesized 4 linear models, in each case averaging the coefficients of the models independently trained on different subsets of the training set.

Applying linear models not to specially generated random subsamples of texts but directly to COHA data by decades, we obtain even higher accuracy (see Table 3). From here on, the short designations of genres used in [20, 21] are adopted: FIC – fiction, MAG – popular magazines, NEWS – newspapers and NF – non-fiction books.

Table 3. Standard deviation of linear estimates of percentage of texts of different genres in the COHA corpus, %.

Model (number of words)	FIC	MAG	NEWS	NF
60,910	0.078	0.096	0.064	0.046
60,635	0.08	0.15	0.14	0.07
6,091	0.29	0.18	0.21	0.07
6,065	0.30	0.19	0.22	0.07

Table 4. Standard deviation of neural network estimates of percentage of texts of different genres in the COHA corpus, %.

Model (number of words)	FIC	MAG	NEWS	NF
60,910	0.97	0.93	1.08	0.52
60,635	0.97	0.57	1.25	0.50
6,091	1.45	1.22	1.14	0.93
6,065	1.61	1.31	1.47	0.99

The standard deviations of the estimates in this case are lower than in the test sample because the corpus size varies over decades by 20–35 million words, and is thus much higher than that of the examples in the test sample.

Neural network models showed slightly lower accuracy on the test sample (standard deviation varied from 2.41 to 2.76). Table 4 shows the standard deviations of neural network estimates based on the COHA corpus data over decades.

Thus, it can be seen that on the COHA data, neural network models are somewhat inferior to linear models in accuracy. Of the four neural network models, the best results were shown by the model with the input vector dimension of 60,635 (the full list of words without abbreviations with a period). Figure 1 shows the estimates of the percentage of the texts of different genres for this model for 15 decades (1860–2009).

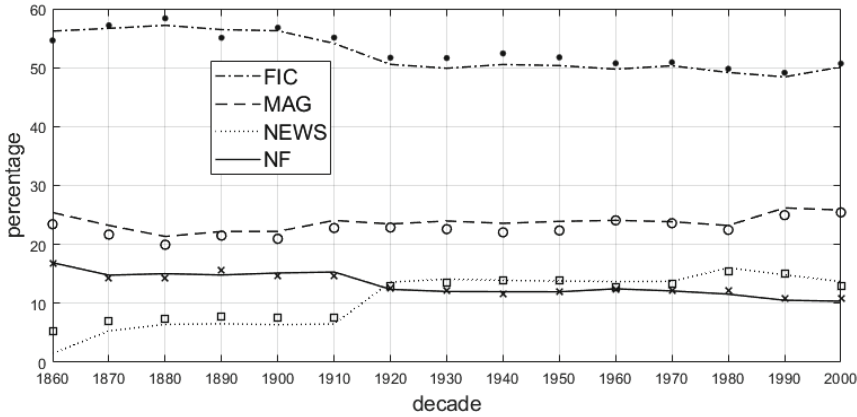


Fig. 1. Percentage of texts of different genres in COHA. The curves show the percentage in accordance with the corpus markup; the markers show the values of the estimates obtained by the neural network model.

3.2 Results for Google Books Ngram

Having trained and tested the models, we applied them to the Google Books Ngram data. Frequency data on the words from the list were extracted from the corpus for each year from 1860–2019 and fed to the neural network input. Figure 2 shows the neural network estimates of the percentage of texts of different genres in the American English subcorpus of GBN. From here on, we show the results for the neural network model with an input vector dimension of 60,635, since it has the lowest range of estimates. Besides the smooth trends, the figure shows a response to such major events as the 1st and 2nd World Wars, as well as significant changes in the genre composition after 2008. The latter may be due to a change in the approach to replenishing the Google Books text collection. Increase in the percentage of fiction texts in the Russian subcorpus of GBN was found in [9], we revealed a similar effect in the English subcorpora.

The actual genre composition of the GBN corpus is unknown. Therefore, one cannot directly find the accuracy of the obtained estimates. However, one can judge the accuracy indirectly, for example, by comparing the readings of several models.

For each curve in Fig. 2, the interval of the most probable values is shown. For this, the standard deviations of the estimates obtained for 15 independently trained models were calculated and plotted on both sides of each curve. It is already clear from Fig. 2 that despite the significantly larger size of the GBN corpus, the range of estimates for the GBN corpus is higher than for the COHA texts. The time-averaged values of the standard deviation of neural network estimates for the four GBN subcorpora are shown in Table 5.

Comparing the values in Tables 4 and 5, one can see that the errors are quite large for the GBN corpus, especially for the percentage of non-fiction texts. The situation is even worse with linear estimates. For the GBN corpus, they lead to artifacts, in particular, in some cases they provide negative percentage values, especially for magazine and current-political texts.

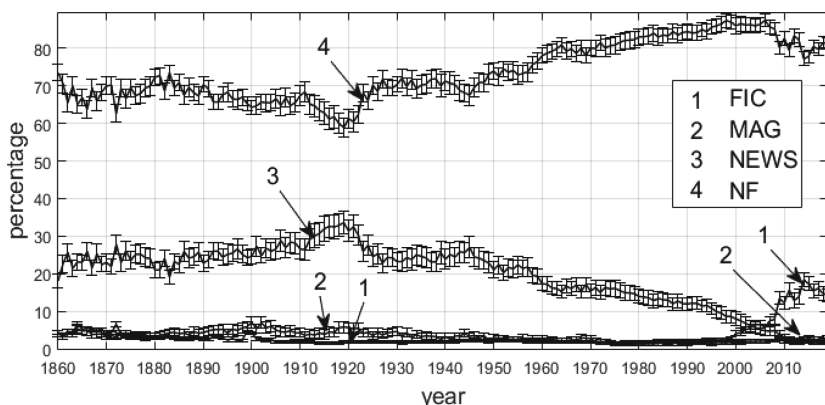


Fig. 2. Percentage of texts of different genres in the American English subcorpus of the Google Books Ngram corpus.

Table 5. Standard deviation of the neural network estimates of the percentage of texts of different genres according to annual data of the Google Books Ngram corpus, %.

Subcorpus	FIC	MAG	NEWS	NF
Common English	0.30	0.64	0.84	2.01
American	0.34	0.85	2.13	2.19
British	0.29	0.63	1.01	1.30
English Fiction	1.20	0.88	0.68	1.40

The question arises, what could be the reason for the decrease of accuracy for the GBN texts? Several assumptions can be made on this account. First, the creators of GBN and COHA used different text tokenization rules. In some cases, this may lead to a shift in the obtained frequencies. Further, the dot was always considered a separator in the GBN version, which, in particular, led to incorrect processing of abbreviations with a dot (abbreviations with a dot were processed correctly in COHA). Although abbreviations with a dot are included in the 3rd version of GBN, it is easy to see that there are many cases when the abbreviation and the dot were counted separately (for example, the word forms ‘Mr.’ and ‘Mrs.’). Apparently, for this reason, we did not obtain good results for the models whose word lists included abbreviations with a dot. Further, the GBN corpus contains book texts, and there are no news items as such. However, books on current political topics have a similar lexical composition, and the model classifies these texts accordingly. However, in the training sample this type of text was represented by a large number of short messages, and books from GBN that are close in topic are still significantly longer. Apparently, this can also be a source of errors. Another important reason may be that the model was trained on American English texts, and 3 out of 4 GBN subcorpora also contain texts in British English.

3.3 Size of English Fiction Texts in Google Books Ngram

Another way to judge the accuracy of the estimates of the genre composition of GBN is to compare the estimates obtained for different English-language subcorpora. A particularly favorable opportunity in this direction is associated with the presence of a separate subcorpus of English Fiction. Let us try to determine how many fiction texts in English are in GBN. It is natural to assume that the size of the corresponding subcorpus (see Table 1) provides an upper estimate for the number of fiction texts. However, as was said above, there is little doubt that this subcorpus also includes some non-fiction texts. Figure 3 shows the percentage estimates of texts of different genres in the English Fiction subcorpus.

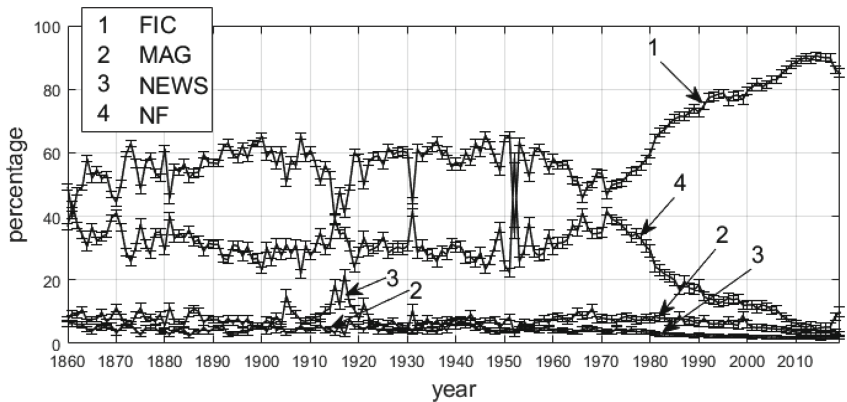


Fig. 3. Percentage of texts of different genres in the English Fiction subcorpus of the Google Books Ngram corpus.

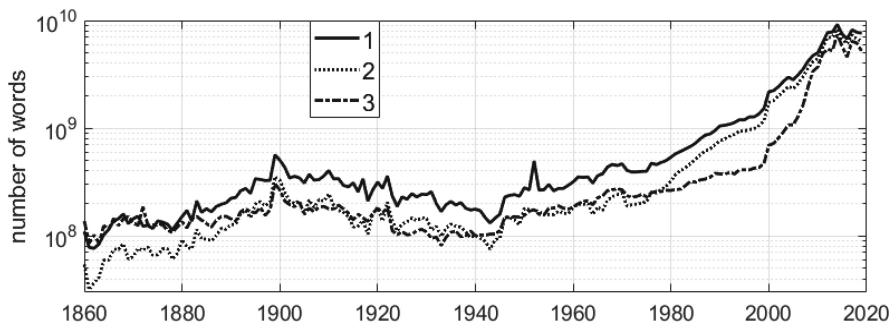


Fig. 4. Size of fiction texts in the Google Books Ngram subcorpora. 1 – size in the English fiction sub-corpus; 2 – estimated size of fiction texts in the English Fiction subcorpus; 3 – estimated size of fiction texts in the English Common subcorpus of Google Books Ngram corpus.

By multiplying the percentage of fiction texts shown in this figure by the size of the subcorpus in a given year, we obtain the first estimate of the size of fiction texts in GBN.

Below in Fig. 4, the corresponding dependence is marked by '2', and the annual size of the English Fiction subcorpus is marked by '1'. An alternative estimate can be obtained either based on the percentage of fiction texts in the Common English subcorpus, or as the sum of the sizes of fiction texts in the American and British English subcorpora. The second option is questionable, since the size of the Common English subcorpus is significantly larger than the sum of the subcorpora of the two national variants of the English language. Thus, the curve marked by '3' in Fig. 4 is obtained by multiplying the percentage of fiction texts in Common English by the size of this corpus in a given year.

Thus, curve '1' gives us an upper bound for the size of fiction texts, and curves '2' and '3' are obtained as a result of employing two methods of calculating the size of such texts. As can be seen, the two curves match well approximately between 1890 and 1980, but there are significant discrepancies over the next three decades. If one sum up the data for the entire time interval 1860–2019, the total size of the English Fiction subcorpus for these years is 155.8 billion words, and the size of the Common English subcorpus is 1907.2 billion words. Thus, the percentage of fiction texts in these years should be no more than 8.17%. Summing up the estimates of the neural network model for the Common English subcorpus for 1860–2019 provides the percentage of fiction texts of 4.99%. Summing up the estimates of the neural network model for the English Fiction subcorpus yields a total size of fiction texts of 120.3 billion words, or 6.31% of the total size of the Common English subcorpus for the specified years. There is a certain discrepancy. Moreover, this discrepancy (by 1.32%) is close in value to the

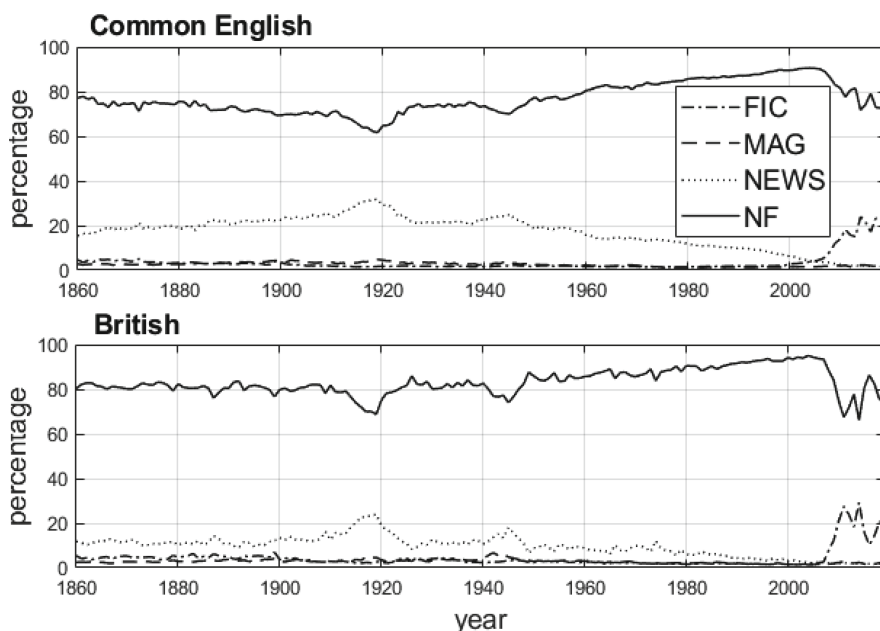


Fig. 5. Percentage of texts of different genres in the Common English (upper figure) and British (lower figure) subcorpora of the Google Books Ngram corpus.

standard deviation of the percentage of fiction texts estimates determined by us above (see Table 5).

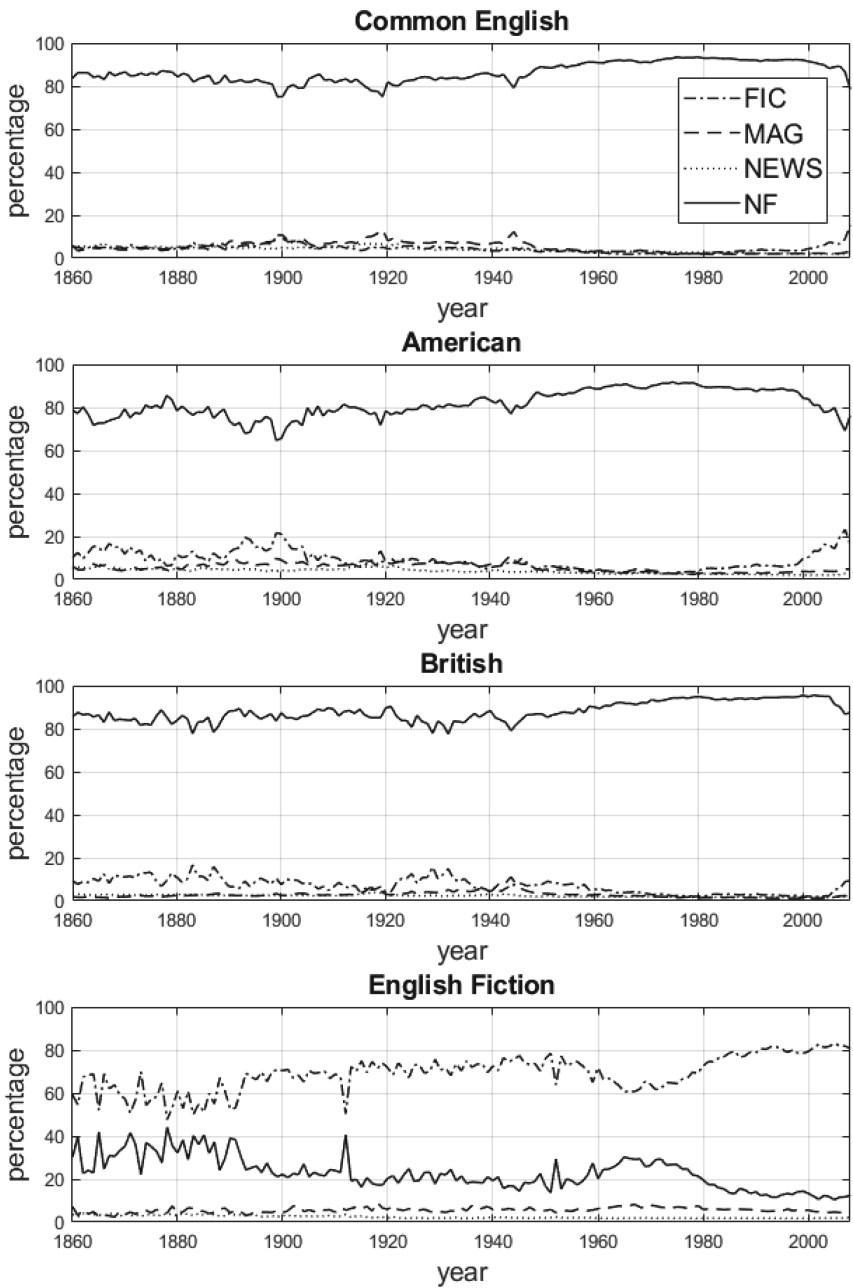


Fig. 6. Percentage of texts of different genres in the subcorpora of the 2nd version of Google Books Ngram. From top to bottom: Common English, American, British, English Fiction.

3.4 Plots for the Other Subcorpora

At the end of this section, we present plots for the two remaining subcorpora of the 3rd version of GBN (Fig. 5), and for all four subcorpora of the 2nd version of GBN (Fig. 6). As can be seen, the nature of the curves for the 2nd and 3rd versions is generally similar, although the specific weight of various genres in these versions somewhat differs. When comparing the graphs, it should be borne in mind that the 2nd version provided data up to and including 2008. Therefore, the segment with large changes in the genre composition after 2008, which is typical of the 3rd version, is missing in Fig. 6.

4 Conclusion

The Google Book Ngram corpus, first published in 2009, has attracted much attention from researchers. Its English language subcorpora including texts of 16.6 million books published since 1470 have a total size of 2 trillion words. Due to its incredible size and coverage of a large time interval, the corpus is widely used in studies of culture, social psychology, and language evolution. However, there are also quite a few critical publications pointing out the limitations of Google Books Ngram. Apparently, the most serious limitation of the corpus is its genre imbalance and lack of information on the genre composition of the books included in the corpus. This seriously complicates researchers' work. At that, the use of existing methods for solving the problem of genre identification is complicated by inaccessibility to the full texts of the Google Books Ngram corpus.

In this paper, we presented an algorithm for estimating the genre composition of the Google Books Ngram corpus. To estimate the percentage of texts of different genres, we use data on relative frequencies for a large range of words (we conducted experiments with word lists including from 6 to 60 thousand words). Both linear models and multilayer feedforward neural networks were tested as predictors. To train the predictors, we used random subsamples of texts from the COHA corpus which are marked up by genres.

Linear predictors provide estimates of the genre composition of COHA texts with very high accuracy, however, cope much worse with Google Books Ngram texts. Neural network predictors that are somewhat inferior to linear ones in accuracy on COHA texts, provide better estimates for Google Books Ngram.

To assess the achieved accuracy, a cross-validation was performed by comparing the estimates of a group of models independently trained on different subsets of the training set. Based on the analysis, it can be assumed that the standard deviation of the neural network estimates obtained from annual data is no worse than 2–2.2%. A comparison of estimates of the size of fiction texts was also performed in two ways using the Common English and English Fiction subcorpora. The best match is obtained for the interval 1890–1980.

In addition to smooth trends, the constructed estimates of the genre composition of Google Books Ngram also show a response to major historical events such as World Wars I and II. It should also be noted that the genre composition has changed significantly since 2008, which may be due to a change in the approach to replenishing the Google Books text collection. The results obtained in the work may be useful for studies of language evolution and cultural changes based on Google Books Ngram data.

Acknowledgments. The work is carried out in accordance with the Strategic Academic Leadership Program “Priority 2030” of the Kazan Federal University of the Government of the Russian Federation.

References

1. Google Books Ngram Viewer. <https://books.google.com/ngrams/>. Accessed 30 June 2025
2. Lin, Y., Michel, J.-B., Aiden, E.L., Orwant, J., Brockman, W., Petrov, S.: Syntactic annotations for the Google Books Ngram corpus. In: Li, H., Lin, C.-Y., Osborne, M., Lee, G.G., Park, J.C. (eds.) 50th Annual Meeting of the Association for Computational Linguistics 2012, Proceedings of the Conference, vol. 2, pp. 238–242. Association for Computational Linguistics, Jeju Island, Korea (2012)
3. Michel, J.-B., Shen, Y.K., Aiden, A.P., Veres, A., Gray, M.K., et al.: Quantitative analysis of culture using millions of digitized books. *Science* **331**(6014), 176–182 (2011)
4. Solovyev, V.: Using the Google Books Ngram corpus to study social evolution. *Soc. Evol. Hist.* **23**(2), 144–164 (2024)
5. Pechenick, E.A., Danforth, C.M., Dodds, P.S.: Characterizing the Google books corpus: strong limits to inferences of socio-cultural and linguistic evolution. *PLoS ONE* **10**(10), e0137041 (2015). <https://doi.org/10.1371/journal.pone.0137041>
6. Belikov, V.I.: What and how can a linguist get from digitized texts? *Siberian J. Philol.* **3**, 17–34 (2016). (in Russian)
7. Koplenig, A.: The impact of lacking metadata for the measurement of cultural and linguistic change using the Google Ngram data sets - reconstructing the composition of the German corpus in times of WWII. *Digit. Scholarsh. Humanit.* **32**(1), 169–188 (2017)
8. Solovyev, V.D., Bochkarev, V.V., Akhtyamova, S.S.: Google Books Ngram: problems of representativeness and data reliability. In: Elizarov, A., Novikov, B., Stupnikov, S. (eds.) *Data Analytics and Management in Data Intensive Domains, DAMDID/RCDL 2019*. CCIS, vol. 1223, pp. 147–162. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-51913-1_10
9. Solovyev, V., Ivleva, A.: How to detect imbalances in the Google Books Ngram corpus? In: Karpov, A., Delić, V. (eds.) *Speech and Computer, SPECOM 2024*. LNCS, vol. 15300, pp. 334–348. Springer, Cham. (2025). https://doi.org/10.1007/978-3-031-78014-1_25
10. Kuzman, T., Ljubešić, N.: Automatic genre identification: a survey. *Lang. Resour. Eval.* **59**(1), 537–570 (2025)
11. Pritsos, D., Stamatatos, E.: Open set evaluation of web genre identification. *Lang. Resour. Eval.* **52**(4), 949–968 (2018)
12. Biber, D., Egbert, J.: Using grammatical features for automatic register identification in an unrestricted corpus of documents from the open web. *J. Res. Des. Stat. Linguist. Commun. Sci.* **2**(1), 3–36 (2015)
13. Priyatam, P.N., Iyengar, S., Perumal, K., Varma, V.: Don’t use a lot when little will do: genre identification using URLs. *Res. Comput. Sci.* **70**, 233–243 (2013)
14. Laippala, V., Kyllönen, R., Egbert, J., Biber, D., Pyysalo, S.: Toward multilingual identification of online registers. In: *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, Turku, Finland, pp. 292–297. Linköping University Electronic Press (2019)
15. Kuzman, T., Pollak, S.: Assessing comparability of genre datasets via cross-lingual and cross-dataset experiments. In: Fišer, D., Erjavec, T. (eds.) *Jezikovne tehnologije in digitalna humanistika: Zbornik conference*, pp. 100–107. Institute of Contemporary History (2022)
16. Lagutina, K.V.: Genre classification of Russian texts based on modern embeddings and rhythm. *Autom. Control. Comput. Sci.* **57**(7), 817–827 (2023)

17. Repo, L., et al.: Beyond the English web: zero-shot cross-lingual and lightweight monolingual classification of registers. In: 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop, pp. 183–191. Association for Computational Linguistics (2021)
18. Kuzman, T., Mozetič, I., Ljubešić, N.: Automatic genre identification for robust enrichment of massive text collections: investigation of classification methods in the era of large language models. *Mach. Learn. Knowl. Extr.* **5**, 1149–1175 (2023). <https://doi.org/10.3390/make5030059>
19. Vajjala, S., Shimangaud, S.: Text classification in the LLM era - where do we stand? arXiv preprint [arXiv:2502.11830](https://arxiv.org/abs/2502.11830) (2025)
20. Corpus of Historical American English. <https://www.english-corpora.org/coha/>. Accessed 30 June 2025
21. Davies, M.: Expanding horizons in historical linguistics with the 400-million word Corpus of Historical American English. *Corpora* **7**(2), 121–157 (2012). <https://doi.org/10.3366/cor.2012.0024>
22. Efron, B.: Nonparametric estimates of standard error: the Jackknife, the bootstrap and other methods. *Get access Arrow. Biometrika* **68**(3), 589–599 (1981). <https://doi.org/10.1093/biomet/68.3.589>
23. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning. Adaptive Computation and Machine Learning*. MIT Press (2016)
24. Abadi, M., et al.: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint, [arXiv:1603.04467](https://arxiv.org/abs/1603.04467) (2016)
25. Chollet, F.: Keras (n.d.). <https://keras.io>. Accessed 30 June 2025

Multimodal Systems



Ensembling Synchronisation-Based and Face-Voice Association Paradigms for Robust Active Speaker Detection in Egocentric Recordings

Jason Clarke^{1(✉)}, Yoshihiko Gotoh¹, and Stefan Goetze^{1,2}

¹ Speech and Hearing (SPandH), School of Computer Science, The University of Sheffield, Sheffield, UK

{jclarke8,y.gotoh}@sheffield.ac.uk

² South Westphalia University of Applied Sciences, Iserlohn, Germany
goetze.stefan@fh-swf.de

Abstract. Audiovisual active speaker detection (ASD) in egocentric recordings is challenged by frequent occlusions, motion blur, and audio interference, which undermine the discernability of temporal synchrony between lip movement and speech. Traditional synchronisation-based systems perform well under clean conditions but degrade sharply in first-person recordings. Conversely, face-voice association (FVA)-based methods forgo synchronisation modelling in favour of cross-modal biometric matching, exhibiting robustness to transient visual corruption but suffering when overlapping speech or front-end segmentation errors occur. In this paper, a simple yet effective ensemble approach is proposed to fuse synchronisation-dependent and synchronisation-agnostic model outputs via weighted averaging, thereby harnessing complementary cues without introducing complex fusion architectures. A refined preprocessing pipeline for the FVA-based component is also introduced to optimise ensemble integration. Experiments on the Ego4D-AVD validation set demonstrate that the ensemble attains 70.2% and 66.7% mean Average Precision (mAP) with TalkNet and Light-ASD backbones, respectively. A qualitative analysis stratified by face image quality and utterance masking prevalence further substantiates the complementary strengths of each component.

Keywords: Face-voice association · Audiovisual active speaker detection · Egocentric recordings

1 Introduction

Audiovisual active speaker detection (ASD) involves identifying the framewise speaking activity of a candidate speaker through the joint analysis of audio signals and temporally aligned face tracks [2, 11, 14, 20, 22, 27, 30]. Traditional ASD systems rely on modelling the temporal correspondence between speech in the audio signal and visual speech-related cues—such as lip movement or cheek posture [11]—in the candidate speaker’s face track. These synchronisation-based

approaches assume audiovisual alignment as a prerequisite for detecting speech activity; this assumption dominates modern methods [22, 24, 30, 33]. Extensions to this framework incorporate contextual cues pertaining to inter-speaker relationships [21, 24] and latent information describing the audible context of each scene [9], these extensions help to address multi-talker scenarios and environmental noise, respectively. However, such methods remain fundamentally contingent on the discernibility of audiovisual synchrony, resulting in these approaches still being vulnerable to the challenges posed in egocentric settings [10, 15].

In egocentric recordings, e.g. captured by head-worn recording devices, such as smart or augmented reality (AR) glasses, synchronisation-based ASD performance deteriorates significantly when compared to their performance on exocentric benchmarks [27]. This is largely attributed to the prevalence of visual occlusions, motion blur, and audio interference from overlapping speech or environmental noise [9, 10, 15, 17, 18, 33], all of which are common challenges in egocentric data. Since synchronisation-based methods require sustained discernable audiovisual cues, these challenges significantly degrade their performance.

To circumvent these limitations, recent work by the authors of this paper has explored using face-voice association (FVA) for the task of ASD, as exemplified by the Self-Lifting for Audiovisual Active Speaker Detection (SL-ASD) architecture [3]. Generally, FVA [7, 25, 28, 34] concerns the task of attributing pre-segmented speaker-invariant utterances to visible identities using cross-modal biometric information rather than temporal alignment. The SL-ASD architecture [3] builds upon this concept by adapting FVA [7] for ASD. This type of approach identifies and leverages transient high-quality facial frames to establish robust voice-face mappings, bypassing the need for fine-grained audiovisual cues being consistently discernable. Prior work [3] has demonstrated robust performance in the context of egocentric recordings achieving mAP scores close to the state-of-the-art despite using significantly less learnable parameters, exclusively for the task of ASD. However, it has been observed [3] that solely relying on face-voice associations introduces two main limitations: face-voice associations falter during speaker-variant utterances (i.e. overlapping speech), and missed speech detections by the speaker-invariant front-end are harshly penalised when the pipeline is evaluated for ASD, holistically. These shortfalls are distinct to the limitations of synchronisation-based methods which struggle more with visual degradation but typically have good recall when the speech signal is audible [9, 10, 19]. By leveraging the complementary strengths of these two paradigms, this work extends the existing SL-ASD approach [3] and proposes a simple yet effective ensemble approach that combines the benefits of synchronisation-agnostic (i.e. FVA-based) and synchronisation-dependent methods of ASD.

More precisely, the presented system integrates two symbiotic components as an ensemble: (i) a synchronisation-based model that captures temporal audiovisual correspondence [22, 30], and (ii) a speaker-invariant segmentation front-end paired with a FVA module, derived from prior work [3] but with refinements for enhanced ensemble performance. The proposed ensemble aggregates output probability sequences from both systems, via weighted averaging, which mitigates each component's divergent failure modes. Although the ensemble mecha-

nism is architecturally lightweight—requiring only a weighted mean fusion of two probability streams—its empirical efficacy demonstrates that synergistic modality insights can be leveraged without complex cross-model attention or gating networks. This simplicity encourages easier deployment on resource-constrained wearable devices.

Contributions:

1. A lightweight late-fusion ensemble method for ASD that combines synchronisation-based and FVA-based models, improving robustness under visual occlusion and audible noise.
2. A refined preprocessing pipeline for SL-ASD to optimise ensemble performance.
3. Empirical validation on Ego4D-AVD: the ensemble achieves 70.2% and 66.7% mAP for two synchronisation-based components (TalkNet and Light-ASD), marking a new state-of-the-art in the domain of egocentric ASD.
4. Qualitative analysis of performance including granular evaluations stratified by Face Image Quality Assessment (FIQA) and randomised utterance masking prevalence to demonstrate the vulnerabilities and strengths of each component of the ensemble.

2 Methodology

This section first provides a brief overview of the typical single-candidate synchronisation-based paradigm used for ASD in Subsect. 2.1, and then describes the FVA-based approach to ASD used by this work in Subsect. 2.2. Finally, the details of the proposed ensemble method, which effectively combines the two synergistic approaches, are presented in Subsect. 2.3.

2.1 Synchronisation-Based Approach to Active Speaker Detection

Conventional single-candidate ASD systems operate by assessing the temporal alignment between cues indicative of speech in a given face track signal and the concurrent audio signal as illustrated in Fig. 1.

A face track $\mathcal{V}_S = \{\mathbf{V}_{S,1}, \dots, \mathbf{V}_{S,T}\}$ is defined as a sequence of T contiguous bounding box face crops $\mathbf{V}_{S,t} \in \mathbb{R}^{H \times W}$ of height H and width W , centred on a single candidate speaker S and the concurrent audio signal is defined as a vector of T_A waveform samples $\mathbf{a} \in \mathbb{R}^{T_A}$ (note that T_A differs from T due to frame rate differences in audio and video modalities).

First, an audio encoder processes the audio signal \mathbf{a} , and a video encoder processes face tracks \mathcal{V}_S , each producing an embedding with shared dimensions. Specifically, the audio branch yields $\mathbf{F}_A \in \mathbb{R}^{T \times d}$ and the video branch yields $\mathbf{F}_V \in \mathbb{R}^{T \times d}$, where d is the embedding dimension of the respective encoders. These two embeddings are then fused to create a single multimodal representation \mathbf{F}_{AV} . Common fusion operations include channel-wise concatenation,

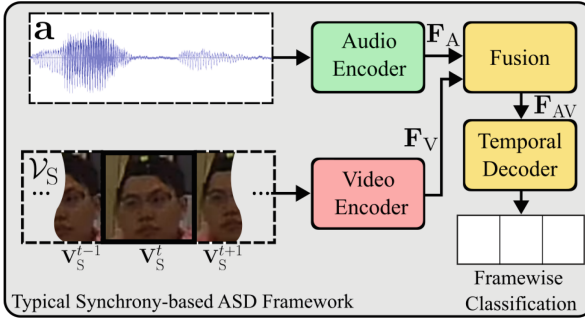


Fig. 1. Typical synchronisation-based single-candidate approach to ASD [22,30].

element-wise summation [22], or attention-based weighting [30]. Regardless, \mathbf{F}_{AV} encodes both audio and visual information at each video-frame.

Finally, a temporal decoder (e.g. a lightweight transformer or temporal convolutional network) is applied along the T dimension of \mathbf{F}_{AV} to model longer-range dependencies in speech activity. A frame-wise classification head then produces probabilities indicating whether the candidate is active at each video-frame. This pipeline—embodied by architectures such as TalkNet [30] and Light-ASD [22]—relies fundamentally on audiovisual synchrony, requiring high-quality lip motion and clean audio to be consistently available for accurate detection.

2.2 Face-Voice Association for Active Speaker Detection

The face-voice association approach to ASD replaces the need for explicit audio-visual synchronisation-based modelling by leveraging cross-modal biometric correspondence. This paper follows prior work, specifically the SL-ASD architecture [3], but deviates in terms of preprocessing implementation which has been optimised by this study for the ensemble approach described in Subsect. 2.3. Hence, the method proposed here will be denoted as SL-ASD†, which is outlined as follows.

Front-End Segmentation and Embedding. Let \mathcal{C} denote the set of video clips in a given dataset. First, an off-the-shelf speaker-diarisation front-end [4] is applied to the audio signal \mathbf{a}_c of each clip c , segmenting each clip into a set of speaker-invariant utterances. Each utterance $\mathbf{u}_{c,i}$ is then embedded by a pretrained speaker-recognition model [12] yielding an embedding $\mathbf{u}'_{c,i} \in \mathbb{R}^{d_s}$ for all utterances, where d_s is the embedding dimension of the speaker recognition model. Collectively, these embeddings form $\mathcal{U}' = \{\mathbf{u}'_{c,i} \mid c \in \mathcal{C}, i = 1, \dots, N_c\}$, where N_c is the number of utterances in clip c . For this segmentation, the Pyanote Audio diarisation model [4] is used because of its robust performance in the task of audio-only diarisation [32].

Additionally, every face-crop image $\mathbf{V}_{S,T}$ in the dataset is embedded by a pretrained face-recognition model [29] yielding a hierarchical set of face-recognition embeddings $\mathcal{X} = \{\mathbf{X}_{c,s} \mid s \in \mathcal{S}_c, c \in \mathcal{C}\}$, where each matrix $\mathbf{X}_{c,s} = [\mathbf{x}_{c,s,1}, \mathbf{x}_{c,s,1}, \dots, \mathbf{x}_{c,s,T_{c,s}}]$ contains face embedding vectors per speaker s in clip c and different $\mathbf{X}_{c,s}$ may be of different size due to variability of frames $T_{c,s}$ per clip and speaker. \mathcal{S}_c is the set of visible identities in clip c , and $T_{c,s}$ is the number of frames for identity s in clip c .

Self-Lifting for Active Speaker Detection. During training, the audio component of each batch consists of several speaker-embeddings $\mathbf{u}'_{c,i}$ sampled from \mathcal{U}' ensuring each utterance was taken from the same clip and spoken by the same identity (as per groundtruth annotation). During inference, since groundtruth annotation for utterance identity is not available, the audio component of each batch is simply a single speaker embedding. For both training and inference, the visual component of each batch comprises $\{\mathbf{X}_{c,s} \mid \forall s \in \mathcal{S}_c\}$, where c refers to the clip from which the speaker embedding(s) in the audio component of the batch were taken from. Each component of the batch is then fed through the relevant branch of the pretrained Self-Lifting [3] model, resulting in $\mathbf{U}'' \in \mathbb{R}^{N_u \times d}$ and $\mathbf{X}'_c \in \mathbb{R}^{|\mathcal{S}_c| \times (\max_{s \in \mathcal{S}_c} T_{c,s}) \times d}$, from the audio and visual branches, respectively. Here, N_u denotes the number of utterances in the audio component of the batch, which is set to 1 during inference. To account for variable visual quality—common in egocentric footage—a lightweight transformer encoder is applied over each sequence dimension (frame dimension) for each visible identity in \mathbf{X}'_c . Through its self-attention mechanism, low-quality frames (e.g. blurred or occluded) are down-weighted, and the resulting sequence is mean-pooled to produce a single quality-aware face-recognition embedding for each identity in the visible component of the batch, resulting in $\mathbf{X}''_c \in \mathbb{R}^{|\mathcal{S}_c| \times 1 \times d}$.

Finally, cross-modal association scores are computed by measuring similarity between the embedded utterance and each aggregated face-recognition embedding in the video component of the processed batch, as illustrated in Fig. 2. Specifically, scaled dot-product cross-attention is used to produce a matching probability that a given utterance was spoken by each visible identity. This pure face–voice association pipeline thus attributes each speech segment to the most likely visible identity, relying only on biometric consistency rather than audio–visual synchrony.

2.3 Ensembling Synchronisation-Based and FVA-based Approaches to Audiovisual Active Speaker Detection

While synchronisation-based (cf. Subsect. 2.1) and FVA-based approaches (cf. Subsect. 2.2) offer complementary strengths, each exhibits vulnerabilities under challenging audiovisual conditions when used in isolation. To mitigate these limitations, an ensemble strategy is employed which fuses predictions from both paradigms by averaging their respective probability sequences.

Let $\mathbf{p}_{\text{sync}} \in [0, 1]^T$ denote the frame-level speaking probabilities predicted by a synchronisation-based model for a given face track. Let $\mathbf{p}_{\text{assoc}} \in [0, 1]^T$

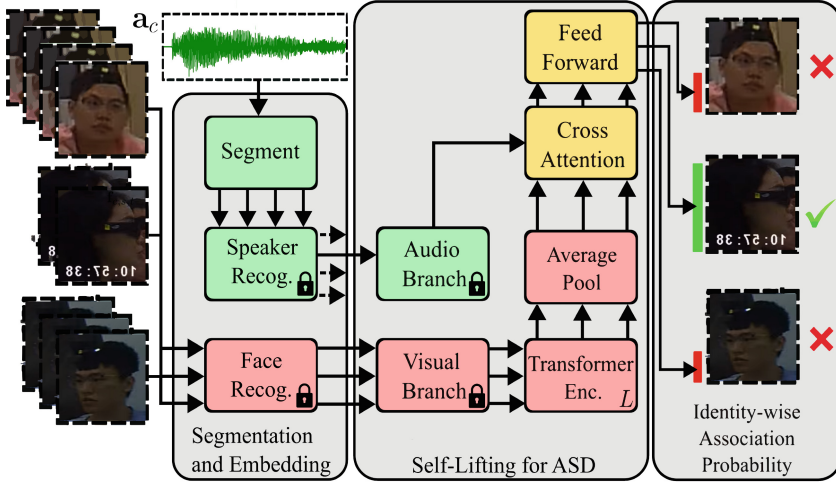


Fig. 2. SL-ASD† framework. Colours indicate modality. Bars adjacent to faces on the right indicate probability of a face-voice match.

denote the probability sequence derived from the FVA-based model for the same hypothesis track as in \mathbf{p}_{sync} . Forming $\mathbf{p}_{\text{assoc}}$ is achieved by projecting the face-voice matching probability uniformly across all concurrent frames in each face track that temporally overlaps with the given utterance. The final ensemble prediction \mathbf{p}_{ens} is then computed via framewise weighted mean averaging, where α is a mixing coefficient determined empirically:

$$\mathbf{p}_{\text{ens}} = \alpha \mathbf{p}_{\text{sync}} + (1 - \alpha) \mathbf{p}_{\text{assoc}}. \quad (1)$$

This late-fusion scheme requires no additional training and yields a probability sequence that integrates both dynamic synchronisation cues and cross-modal biometric consistency. The resulting ensemble consistently outperforms either constituent method when used in isolation, particularly in scenarios with degraded visual quality or non-frontal faces (cf. Subsect. 4.1).

3 Experiments

This section briefly introduces the egocentric Ego4D dataset used in this work in Subsect. 3.1, the implementation details in Subsect. 3.2, and the evaluation metrics used throughout in Subsect. 3.3.

3.1 Ego4D Dataset for Egocentric Audiovisual Diarisation

The Ego4D dataset [15] comprises egocentric video recordings, totalling 572 unique clips each lasting five minutes in duration, some of which were captured simultaneously. The data was obtained using various wearable devices using

1080p video. The audio signals are standardised to a single-channel in 16 kHz format. Video-frames were recorded at 30 Hz. The dataset reflects real-world conditions – featuring fluctuating lighting, frequent occlusions, and continuously changing viewpoints – making it a particularly demanding testing scenario for ASD. Ego4D-AVD is divided into three non-overlapping folds: 379 clips for training, 50 for validation, and 133 for testing. Because test labels are withheld, the original training set was further split by this work into 110 clips for model training and 23 for development, preserving the reserved validation set for final evaluation. Splits were created to ensure that no individual appears in more than one fold.

3.2 Implementation Details

Synchronisation-Based Models. For this component of the ensemble two different ASD systems were used as baselines, namely TalkNet [30] and Light-ASD [22]. These architectures were implemented under the exact configurations and hyperparameters specified in their original manuscripts apart from the training duration. Each model was trained independently 10 times for 30 epochs, and the checkpoint achieving the best performance on the development-set of Ego4D was selected. Finally, the selected checkpoints were employed to generate the synchronisation-based predictions incorporated into the ensemble.

Self-Lifting for Audiovisual Active Speaker Detection. The SL-ASD[†] implementation was similar to that described in [3]. Specifically, the front-end utterance segmentation was performed on a clipwise basis using the Pyannote.audio-speaker-diarization-3.1 system [4] to extract speaker-invariant utterances. Speaker-recognition embeddings were obtained from these utterances using the ECAPA-TDNN [12] model, pretrained on VoxCeleb2 [8]. Face-recognition embeddings were extracted from all face-track frames in the dataset via Inception-V1 [29] pretrained on VGG-Face2 [5]. For finetuning of the Self-Lifting audio and video encoder branches, the model was instantiated with the implementation described in its original manuscript [7], except the number of cluster centroids, which was reduced to 50 to better reflect the number of distinct identities present in Ego4D. In the ASD adaptation (SL-ASD [3]), all original framework parameters were frozen, and only the transformer encoder, the cross-attention module, and the feed-forward layer were trained explicitly for ASD (cf. Fig. 2). During training, each batch’s audio component comprised all utterances for a single clipwise identity, while its video component included all face-track frames for every visible identity in the clip; during validation and inference, the audio component was limited to single utterances. Optimisation was carried out using Adam with an initial learning rate of 1×10^{-5} , decayed by a factor of 0.2 every 5 epochs, and a single transformer layer with four attention heads was employed for both the encoder and cross-attention.

Face Quality Assessment. To perform a granular evaluation of the various approaches to ASD considered by this work (cf. Subsect. 4.2), a method to quantify the visual quality of the frames in each face-track was employed. In analogy to the well-established domain of Face Image Quality Assessment (FIQA) [16, 23, 31], the per-frame recognisability of the candidate speaker was inferred via the confidence score produced by the pretrained Multi-task Cascaded Convolutional Neural Network (MTCNN) face detector [35]. Specifically, every cropped face image in a groundtruth track was passed through MTCNN, and the resulting detection probabilities were recorded. These per-frame scores were then averaged to yield a single, track-level quality metric.

3.3 Evaluation Metrics

For holistic evaluation, each system is evaluated for ASD using the Cartucho object detection mAP metric [6], which is in alignment with the mAP protocol established by the PASCAL VOC2012 challenge [13]. This evaluation strategy is consistent with the framework adopted by the Ego4D audiovisual diarization challenge [15] and is widely employed in recent ASD research [9, 10]. Owing to the absence of ground truth annotations for the test folds in Ego4D, all results are reported on its validation folds, in accordance with prevailing conventions in the literature [1, 2, 9, 20, 24, 33]. The validation fold is exclusively reserved for testing purposes and are not used during model development. For the evaluations presented in Subsect. 4.2, the problem is reformulated as a binary classification task, with metrics computed using the scikit-learn [26] implementation of average precision.

4 Results

4.1 Comparison with State-of-the-Art Methods

To assess the efficacy of the proposed ensemble, its performance is evaluated holistically against leading ASD systems. Table 1 summarises mAP and parameter counts for each method on the validation fold of the Ego4D-AVD benchmark.

When fused via weighted averaging, the synchronisation-based TalkNet model in conjunction with the face-voice association-based SL-ASD† model yield a combined mAP of 70.2%, outperforming both individual baselines (TalkNet: 51.0%; SL-ASD: 60.7%) by a substantial margin. Crucially, this gain cannot be attributed merely to increased model capacity. This is illustrated by comparing the performance of an ensemble of two synchronisation-based approaches (TalkNet + Light-ASD) of 64.1% with that of Light-ASD + SL-ASD† of 66.7%. While the former yields a significant improvement over its respective baselines, it still exhibits weaker performance than the latter, despite requiring significantly more learnable parameters. This indicates that combining synchronisation-based approaches with FVA-based approaches leverages truly complementary cues.

Moreover, the TalkNet + SL-ASD† ensemble establishes a new state of the art, surpassing the recent LoCoNet [33] model by 1.8% absolute mAP while

Table 1. Comparison with state-of-the-art ASD systems on the validation fold of Ego4D. “ASD Params. [M]” denotes the number of learnable-parameters each system uses exclusively for the task of ASD. All values are taken from published literature except ensemble approaches. SL-ASD[†] indicates the modified implementation of SL-ASD [3]

Model	Ensemble	mAP [%]	ASD Params. [M]
TalkNet [15]	✗	51.0	15.1
Light ASD [10]	✗	54.3	1.0
SL-ASD [3]	✗	59.7	0.4
SPELL [18]	✗	60.7	> 22.5
LoCoNet [33]	✗	68.4	33.5
Light ASD + TalkNet	✓	64.1	16.1
Light ASD + SL-ASD [†]	✓	67.1	1.4
TalkNet + SL-ASD[†]	✓	70.2	15.5

using fewer than half of its learnable parameters exclusively dedicated to ASD. This demonstrates that simple late fusion of heterogeneous ASD paradigms can yield superior accuracy-efficiency trade-offs compared to monolithic architectures, even those that effectively leverage contextual information.

4.2 Qualitative Analysis

To further investigate the hypothesis that FVA-based models leverage information complementary to that of synchronisation-based approaches, a stratified evaluation was conducted. Face tracks were grouped into discrete bins based on

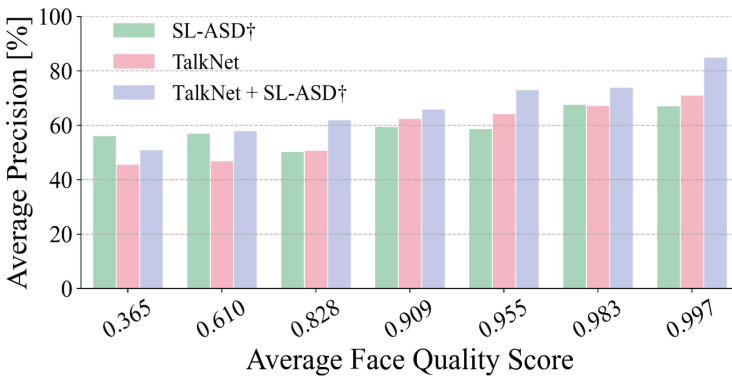


Fig. 3. Comparison of synchronisation-based (TalkNet [30] pink bar), FVA-based (SL-ASD [3], green bar), and ensemble-based (blue bar) approaches to ASD, evaluated on strata of equal size (each comprising tracks with similar average face quality scores). Lower face quality scores indicate tracks with greater visual distortion or occlusion. Irregular face quality score incrementation is due to a non-uniform distribution of trackwise visual quality.

their average face quality scores, enabling a detailed analysis of model performance under varying degrees of visual degradation, including factors such as blur, occlusion, and suboptimal lighting conditions.

The results of this evaluation, shown in Fig. 3, reveal that synchronisation-based models, as speculated [3, 9, 10], exhibit a significant decline in performance as face quality deteriorates. This sensitivity is attributed to their reliance on precise visual cues—particularly lip movements and cheek posture [11]—that must be consistently discernible throughout the duration of the face track. In contrast, the FVA-based model, SL-ASD†, demonstrates a more stable performance across all quality bins. Its robustness stems from the ability to identify and utilise even a limited number of high-quality frames within a sequence. The transformer encoder within SL-ASD† effectively down-weights low-quality frames and emphasizes those that are most informative for identity recognition. This mechanism allows the model to maintain reliable speaker attribution despite transient visual distortions.

Conversely, Fig. 4 conveys the effect of audio degradation on each approach. As the probability of randomised utterance masking is increased, only a modest reduction in average precision is exhibited by the synchronisation-based model, owing to its ability to leverage cross-modal information, in this case video, when the audio is obscured. By contrast, a steeper decline is observed for the face–voice association-based SL-ASD†, since uninterrupted utterance segments are required by its speaker-invariant front-end for robust speaker embedding extraction. Crucially, higher overall performance across all masking levels is maintained by the ensemble approach, which leverages both streams to compensate for audio distortions that would otherwise impair face–voice association. These findings further substantiate that synchrony-dependent and synchrony-agnostic paradigms leverage complementary information for ASD.

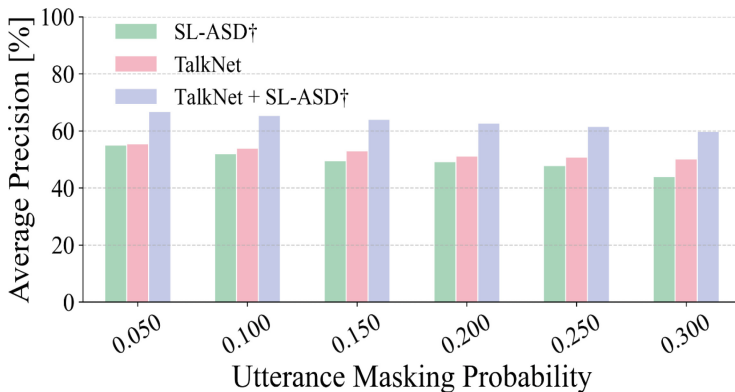


Fig. 4. Comparison of three approaches to ASD on the Ego4D validation set: synchronisation-based (TalkNet [30], pink bar), FVA-based (SL-ASD [3], green bar), and ensemble-based (blue bar) methods. The evaluation is performed with randomised masking applied specifically to utterance regions within the audio signals, simulating various levels of audio signal degradation.

5 Conclusion

In this work, a lightweight late-fusion ensemble for ASD was proposed, combining synchronisation-based and FVA-based models to enhance robustness under visual occlusion and audio interference. The preprocessing pipeline of SL-ASD was refined to optimise its integration within the ensemble, leading to consistent performance gains. Empirical validation on the Ego4D-AVD validation set demonstrated that the ensemble attains 70.2% and 66.7% mAP when paired with TalkNet and Light-ASD backbones, respectively—establishing a new state-of-the-art in ASD. Finally, a qualitative analysis stratified by face quality and utterance masking prevalence was conducted, revealing the complementary strengths and failure modes of each model component. Collectively, these findings substantiate that simple yet principled fusion of synchrony-dependent and synchrony-agnostic streams can reliably mitigate modality-specific degradations in challenging egocentric scenarios.

Acknowledgments. This work was supported by the Centre for Doctoral Training in Speech and Language Technologies (SLT) and their Applications funded by UKRI [grant number EP/S023062/1]. This work was also funded in part by Meta.

References

1. Alcazar, J.L., Cordes, M., Zhao, C., Ghanem, B.: End-to-End active speaker detection. In: European Conference on Computer Vision (2022)
2. Alcazar, J.L., et al.: Active speakers in context. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
3. Authors of this paper (author names redacted, will be added in final version of this paper): Face-voice association for audiovisual active speaker detection in egocentric recordings. In: Submitted to European Signal Processing Conference (EUSIPCO) (2025)
4. Bredin, H., et al.: pyannote.audio: neural building blocks for speaker diarization. In: ICASSP 2020, IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. 7124–7128 (2020). <https://doi.org/10.1109/ICASSP40776.2020.9054260>
5. Cao, Q., Shen, L., Xie, W., Parkhi, O.M., Zisserman, A.: Vggface2: a dataset for recognising faces across pose and age. In: 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018), pp. 67–74. IEEE Press (2018). <https://doi.org/10.1109/FG.2018.00020>
6. Cartucho, J., Ventura, R., Veloso, M.: Robust object recognition through symbiotic deep learning in mobile robots. In: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (2018)
7. Chen, G., Zhang, D., Liu, T., Du, X.: Self-lifting: a novel framework for unsupervised voice-face association learning. In: Proceedings of the 2022 International Conference on Multimedia Retrieval, ICMR '22, pp. 527–535. Association for Computing Machinery, New York (2022). <https://doi.org/10.1145/3512527.3531364>
8. Chung, J.S., Nagrani, A., Zisserman, A.: Voxceleb2: deep speaker recognition. In: Interspeech 2018. ISCA (2018) <https://doi.org/10.21437/Interspeech.2018-1929>

9. Clarke, J., Gotoh, Y., Goetze, S.: Improving audiovisual active speaker detection in egocentric recordings with the data-efficient image transformer. In: IEEE Automatic Speech Recognition and Understanding Workshop (ASRU23) (2023). <https://doi.org/10.1109/ASRU57964.2023.10389764>
10. Clarke, J., Gotoh, Y., Goetze, S.: Speaker embedding informed audiovisual active speaker detection for egocentric recordings. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2025). <https://arxiv.org/abs/2502.06012>
11. Datta, G., Etchart, T., Yadav, V., Hedau, V., Natarajan, P., Chang, S.F.: ASD-transformer: efficient active speaker detection using self and multimodal transformers. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2022). <https://doi.org/10.1109/ICASSP43922.2022.9746991>
12. Desplanques, B., Thienpondt, J., Demuynck, K.: ECAPA-TDNN: emphasized channel attention, propagation and aggregation in TDNN based speaker verification. In: Interspeech 2020. ISCA (2020). <https://doi.org/10.21437/interspeech.2020-2650>
13. Everingham, M., Van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results (2012). <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>
14. Everingham, M., Sivic, J., Zisserman, A.: Hello! My name is... Buffy – automatic naming of characters in TV video. In: British Machine Vision Conference (2006)
15. Grauman, K., et al.: Ego4D: around the world in 3,000 hours of egocentric video. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
16. Hernandez-Ortega, J., Galbally, J., Fierrez, J., Haraksim, R., Beslay, L.: Faceqnet: quality assessment for face recognition based on deep learning. In: 2019 International Conference on Biometrics (ICB), pp. 1–8 (2019). <https://doi.org/10.1109/ICB45273.2019.8987255>
17. Huh, J., et al.: Advancing active speaker detection for egocentric videos. In: ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 1–5 (2025). <https://doi.org/10.1109/ICASSP49660.2025.10888166>
18. Ishibashi, T., Ono, K., Kugo, N., Sato, Y.: Technical Report for Ego4D Long Term Action Anticipation Challenge 2023 (2023). <https://arxiv.org/abs/2307.01467>
19. Jiang, Y., Tao, R., Pan, Z., Li, H.: Target active speaker detection with audio-visual cues. In: Proceedings of Interspeech (2023)
20. Köpüklü, O., Taseska, M., Rigoll, G.: How to design a three-stage architecture for audio-visual active speaker detection in the wild. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021). <https://doi.org/10.1109/ICCV48922.2021.00123>
21. Le'on-Alc'azar, J., Heilbron, F.C., Thabet, A.K., Ghanem, B.: MAAS: multi-modal assignation for active speaker detection. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
22. Liao, J., Duan, H., Feng, K., Zhao, W., Yang, Y., Chen, L.: A light weight model for active speaker detection. In: Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
23. Meng, Q., Zhao, S., Huang, Z., Zhou, F.: Magface: a universal representation for face recognition and quality assessment. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 14220–14229 (2021). <https://doi.org/10.1109/CVPR46437.2021.01400>

24. Min, K., Roy, S., Tripathi, S., Guha, T., Majumdar, S.: Learning long-term spatial-temporal graphs for active speaker detection. In: European Conference on Computer Vision (2022)
25. Ning, H., Zheng, X., Lu, X., Yuan, Y.: Disentangled representation learning for cross-modal biometric matching. *IEEE Trans. Multimedia* **24**, 1763–1774 (2022). <https://doi.org/10.1109/TMM.2021.3071243>
26. Pedregosa, F., et al.: Scikit-learn: machine learning in python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
27. Roth, J., et al.: Ava active speaker: an audio-visual dataset for active speaker detection. In: Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP) (2020). <https://doi.org/10.1109/ICASSP40776.2020.9053900>
28. Saeed, M.S., et al.: Single-branch network for multimodal training. In: ICASSP 2023–2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE (2023)
29. Szegedy, C., et al.: Going deeper with convolutions. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–9 (2014). <https://api.semanticscholar.org/CorpusID:206592484>
30. Tao, R., et al.: Is someone speaking? Exploring long-term temporal features for audio-visual active speaker detection. In: Proceedings of 29th ACM International Conference on Multimedia (2021)
31. Terhörst, P., Kolf, J.N., Damer, N., Kirchbuchner, F., Kuijper, A.: Ser-fiq: unsupervised estimation of face image quality based on stochastic embedding robustness. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5650–5659 (2020). <https://doi.org/10.1109/CVPR42600.2020.00569>
32. Wang, J., Chen, G., Zheng, Y.D., Lu, T.: Exploring detection-based method for speaker diarization @ ego4d audio-only diarization challenge 2022 (2022). <https://arxiv.org/abs/2211.08708>
33. Wang, X., Cheng, F., Bertasius, G.: LoCoNet: long-short context network for active speaker detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2024)
34. Wen, P., Xu, Q., Jiang, Y., Yang, Z., He, Y., Huang, Q.: Seeking the shape of sound: an adaptive framework for learning voice-face association. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16347–16356 (2021)
35. Zhang, K., Zhang, Z., Li, Z., Qiao, Y.: Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Process. Lett.* **23**(10), 1499–1503 (2016). <https://doi.org/10.1109/LSP.2016.2603342>



Phonetic and Visual Characteristics of Cognitive Load

Vera Evdokimova^(✉)  and Maria Maksimova 

Saint Petersburg University, Saint Petersburg, Russia

{v.evdokimova,m.r.maksimova}@spbu.ru

Abstract. The study of speech in different emotional, psychophysiological and cognitive states is an important task for the development of speech systems. Cognitive load is the load on a person's cognitive system when performing a task. This paper analyses speech characteristics and facial features that can serve as the markers of cognitive load. Previous research revealed that cognitive load is associated with increasing fundamental frequency (F0), laryngealization, narrowing F0 range, changing articulation rate. Cognitive load can also be recognized using head pose, eye gaze and facial expressions. Two experiments were conducted in order to study speech and facial movements under cognitive load. During the first experiment, the participants played a driving simulator game and answered general knowledge questions simultaneously. Audio and videosamples were recorded. The information about action units (facial muscle movements) was obtained using Open Face 2.2.0. The results revealed that the most frequent visual characteristics of cognitive load are turning eyes to the right (AU62) and dimpler (AU14, the contraction of the buccinator muscle). In the second experiment, three episodes of a talk show were studied. The interviewer was driving a vehicle and conducting an interview as a dual task. The results showed that the most common visual markers of cognitive load are AU01 (inner brow raising), AU02 (outer brow raising), AU05 (upper lid raiser), AU10 (upper lip raiser), AU15 (lip corner depressor). The findings in both experiments suggest that cognitive load could be recognized by movements in the eye area and lip area.

Keywords: Phonetics · Speech Acoustics · Phonetic and Visual Markers of Cognitive Load · Facial Action Coding System

1 Background

1.1 Cognitive Load and Working Memory

Detecting cognitive load via speech and visual characteristics in a non-invasive way is an essential task in modern times, when people are overwhelmed with various kinds of information on a regular basis. Cognitive load might lead to particularly adverse consequences in human-operated systems (such as vehicles, call centres or air traffic control centres). These problems can be tackled by the automatic recognition of cognitive load and appropriate adjustment to the user's physical and emotional condition. For example, an automatic system may simplify output data for a user's request or paraphrase instructions for a certain procedure in less complex terms.

One of the most perspective ways to detect the presence of the cognitive load is testing the speech and facial expressions when speaking. It is possible to communicate with the operators or the drivers and detect the change in speech characteristics and simultaneously mimics. The constant presence of the phones with cameras makes it an easy task.

According to J. Sweller, the founder of cognitive load theory, cognitive load refers to the load on a human's cognitive system during task performance. It also represents the amount of data that needs to be simultaneously held in working memory in order to complete a task [1, 2].

Working memory (also known as short-term memory) is responsible for the temporary storage and processing of information. The main component of working memory is 'central executive' that regulates the rest of working memory constituents [3] (Fig. 1).

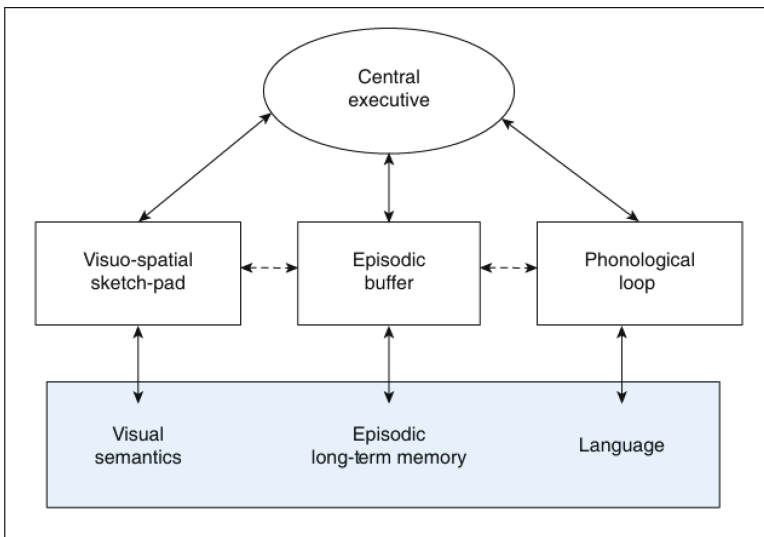


Fig. 1. The model of working memory [3].

Visuo-spatial sketchpad is the part of working memory which process visual and spatial stimuli. Visual data implies objects' appearance (e.g., colours, sizes, shapes) whereas spatial information refers to an object location in relation to other items and with navigation (spatial orientation) [3].

Acoustic and verbal data is stored in and processed by phonological loop. This working memory component can keep information only for short periods of time, but rehearsing sound sequences mentally helps to hold them in storage [3].

The fourth part of working memory is called 'episodic buffer'. Its function is storing multidimensional information related to various senses (such as visual and sound information, taste, smell). Episodic buffer also enables interaction between several components of working memory, each of which process data from different sources.

Since working memory has a limited processing capacity, it is essential to control cognitive load level in order to perform a task effectively [3].

The Types of Cognitive Load. J. Sweller identifies the following types of cognitive load: intrinsic, extraneous and germane [1, 3].

Intrinsic cognitive load depends on the complexity of the task which is determined by the level of element interactivity. Intrinsic cognitive load is also influenced by the learner's background knowledge and experience.

Extraneous cognitive load is not related directly to the complexity of the task. This type of cognitive load is concerned with the method of presenting information to learners and instructing them how to complete the task.

Germane (effective) cognitive load is closely connected to intrinsic load. The function of germane load is directing cognitive resources to activities relevant to the task so that a learner is not distracted by extraneous activities. However, extraneous load (the presentation of learning material and instructional procedures) can affect germane load. This type of load is also related to motivation. If an increase in motivation is directly linked to learning, it leads to an increase in germane cognitive load. In contrast to the other types of cognitive load, augmenting germane load has a favourable impact on learning.

The present research focuses on germane cognitive load because the experiment consists of the main task (driving using a simulator) and a dual task (answering questions).

1.2 The Speech Markers of Cognitive Load

There are different methods of measuring cognitive load, including the analysis of voice and speech.

The increase of cognitive load leads to a higher rate of opening and closing of vocal folds. As a consequence, subglottal pressure rises and, therefore, fundamental frequency (F0) increases. Cognitive load can also be characterized by laryngealization (creaky voice). Regarding formant values, previous research did not reveal any regularities in their changes under cognitive load [4, 5].

Huttunen et al. [6] conducted a research on the cognitive load in military pilots' speech during a training session in a flight simulator. The findings showed that cognitive load can be accompanied by increasing mean F0, narrowing F0 range and increasing voice intensity (loudness). Yap [5] revealed that cognitive load can be marked by different parameters depending on the level of cognitive load. For example, high cognitive load is associated with a higher speech tempo, while medium and low levels do not generally cause speech tempo to change significantly.

Berthold and Jameson [7] defined cognitive load as the effort made in order to perform the main task and an additional task simultaneously. The participants of their experiment were asked to solve the problem with the car by talking to a mechanic on the phone. The primary task was communicating with a mechanic. Simultaneously, the subjects were repairing the car, looking for the necessary tools, talking to other people.

In the introduction to their paper, Berthold and Jameson summarize phonetic markers of cognitive load mentioned in previous works. It is reported that the number and duration of pauses increases and articulation rate (as well as speech tempo) lowers under cognitive load. The cognitive load recognition system presented by Berthold and Jameson is based

on the features mentioned. Müller et al. [8] studied the effect of cognitive load on articulation rate under various conditions (namely, time pressure and multitasking). The main task for the participants of their experiment was to ask questions to an electronic assistant. The questions were related to the images appearing on the screen. A dual task was to navigate in a simulator. It was found that time pressure can lead to an increase of articulation rate, whereas dual task performing can result in a lower articulation rate. Thus, it can be concluded that speech under cognitive load can be characterized by the following features:

1. Increased F0
2. Narrowed F0 range
3. Increased intensity
4. Laryngealization
5. Formant shifts
6. Speech tempo and articulation rate variations.

1.3 Visual Characteristics of Cognitive Load

Research on visual markers of cognitive load is essential as cognitive load can be associated with head pose, eye movements and facial expressions.

As stated in Khavylo et al. [9], the micro-movements of facial muscles might serve as indicators of cognitive load. During the experiment, the subjects were asked 100 questions in total. Half of the questions were labeled as ‘complex’ and another half as ‘simple’. The experiment sessions were filmed. A model for evaluating question complexity was developed. To determine complexity, the following variables were used for every question and for every muscle: the time of thinking, facial movement variance before and after the answer, the time of answering. The average accuracy of 95,7% was achieved. This suggests that facial muscle movements can be reliable markers of cognitive load.

The previous research also detected changes in the amplitude of muscle movements before answering which were not due to articulation processes. These periods of mimic activity before answers are considered to be indicators of cognitive load [10].

For describing and classifying facial expressions, FACS (Facial Action Coding System) is widely used [11]. This system allows to describe facial expressions in terms of action units. Action units denote facial muscle movements that form a certain facial expression. FACS is utilized for the recognition of human’s mental and physical condition (in particular, for cognitive load detection).

Action units extraction from visual data can be performed using OpenFace [12]. This tool is designed for facial expression analysis. It enables detecting facial landmarks, recognizing action units, tracking head pose and eye gaze.

There is some previous research on using visual data to enhance speech recognition. Ivanko et al. [13] developed the RUSAVIC (Russian AudioVisual Speech in Cars) dataset. It contains speech samples and audio recordings. The recordings were taken in various noise levels. The participants were asked to pronounce the most frequent commands to an in-car voice assistant. They also read aloud letters of the Russian alphabet and digits.

Ivanko et al. [13] state that drivers’ speech recognition is essential because using hands to operate navigation system might lead to distraction and, consequently, to traffic

accidents. However, detecting drivers’ physiological and mental state (including cognitive load) is also a vitally important task. The probability of human error, which is a common cause of car accidents, rises with decreasing cognitive abilities. Cognitive resources in turn can be affected by stress, fatigue and other psychological and physical conditions.

2 The Preliminary Experiment (Stroop Test)

2.1 Materials and Methods

The aim of the preliminary experiment was to compare speech under cognitive load and without cognitive load.

The Stroop test [14] was used as a main task causing cognitive load. This is a psychological test which allows to evaluate an individual’s ability to process two stimuli presented simultaneously. The reaction to one stimulus should be inhibited if it impedes the processing of another stimulus [15].

Seven female students aged from 20 to 25 were presented words denoting colour names in which the font colour did not match the word meaning. The primary task was to read aloud the font colour name. Therefore, the ability to process verbal information (reading the word and understanding its meaning) was inhibited in order to process visual stimuli. As a dual task, the subjects answered questions related to their field of studies. The speech samples were recorded at the recording studio.

To record speech without cognitive load, participants were asked to read aloud a phonetically representative text. The total duration of the recordings was 49 min 42 s.

The F0 values and the duration values of vowels, consonants (sonorants and fricatives) and laryngealized segments were calculated using scripts from the SpeCT (The Speech Corpus Toolkit for Praat). The duration values obtained were rounded to the nearest whole number.

2.2 Results

The results of the Stroop test experiment are presented in Tables 1, 2, 3 and 4. The sounds with the most noticeable differences in duration under the two conditions are listed in the following tables.

Table 1. The duration values of stressed vowels under cognitive load and without cognitive load.

	Under cognitive load (ms)	Without cognitive load (ms)
/i/	87	67
/a/	119	82
/o/	96	51

Table 2. The duration values of unstressed vowels under cognitive load and without cognitive load.

	Under cognitive load (ms)	Without cognitive load (ms)
/i/	132	78
/a/	178	68
/i/	331	65

Table 3. The duration values of sonorants under cognitive load and without cognitive load.

	Under cognitive load (ms)	Without cognitive load (ms)
/m/	119	99
/n/	207	52

Table 4. The duration values of fricatives under cognitive load and without cognitive load.

	Under cognitive load (ms)	Without cognitive load (ms)
/s/	201	89
/s'/	324	146
/ž/	175	71

It was revealed that the duration of unstressed vowels was greater under cognitive load by 120 ms on average. However, these differences were statistically significant solely for the /i/ vowel in post-nuclear parts of phonetic words. The duration values of stressed vowels were also higher under cognitive load (on average, by 36 ms). As in the case of unstressed vowels, the results were not always statistically significant. A significant difference was found only regarding the /a/ vowel.

Sonorants and fricative consonants also had a longer duration under cognitive load. On average, the duration of fricatives was by 120 ms greater when the speakers experienced cognitive difficulties. The duration values of sonorants were higher by 85 ms, on average. Nevertheless, these results were not statistically significant.

The duration of laryngealized segments was also greater under cognitive load (on average, by 115 ms).

Regarding F0, the participants frequently produced rising or falling-rising intonation. The F0 range was, on average, by 4 semitones narrower under cognitive load than in reading non-final phrases in the text.

3 The Driving Simulator Experiment

3.1 Materials

The purpose of the driving simulator experiment was to study both acoustic and visual markers of cognitive load as well as their interplay.

Sixteen subjects participated in this experiment, six male and ten female. The participants were aged from 19 to 52, with an average age of 33.

The subjects played a driving simulator game (“City Car Driving” [16]) on a computer and simultaneously answered general knowledge questions. Some of the questions required listing of certain objects as a response (e.g., “What cities that start with the letter ‘R’ do you know?”), while others implied a detailed answer (“Describe the water cycle”).

In total, sixteen video fragments were recorded using a smartphone (one recording for each participant) with FullHD 1920x1080 screen resolution (mp4 format). The duration of the recordings varied from 7 to 13 min. The total duration of the material was 2 h 46 min.

3.2 Methods

The videos obtained were processed using OpenFace [12] (version 2.2.0). The frequency and intensity values of action units were extracted from the recordings. Facial gestures in the videos were also manually annotated using ELAN as the face area was not properly detected by OpenFace in some cases. The sound files were transcribed automatically using WEBMAUS Basic. The resulting orthographic and phonetic transcriptions were corrected manually using Praat. Acoustic parameters were obtained using SpeCT (The Speech Corpus Toolkit for Praat) scripts. The duration values of sounds were rounded to the nearest whole number (Figs. 2, 3 and 4).

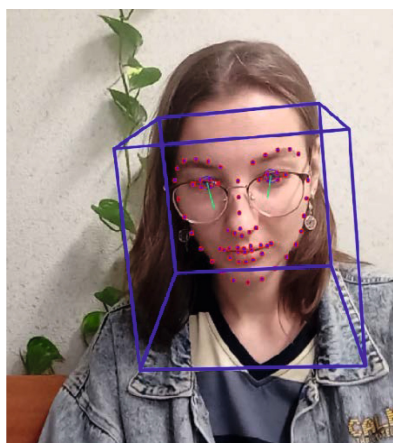


Fig. 2. The visualization of facial landmarks, eye gaze and head pose using OpenFace 2.2.0.

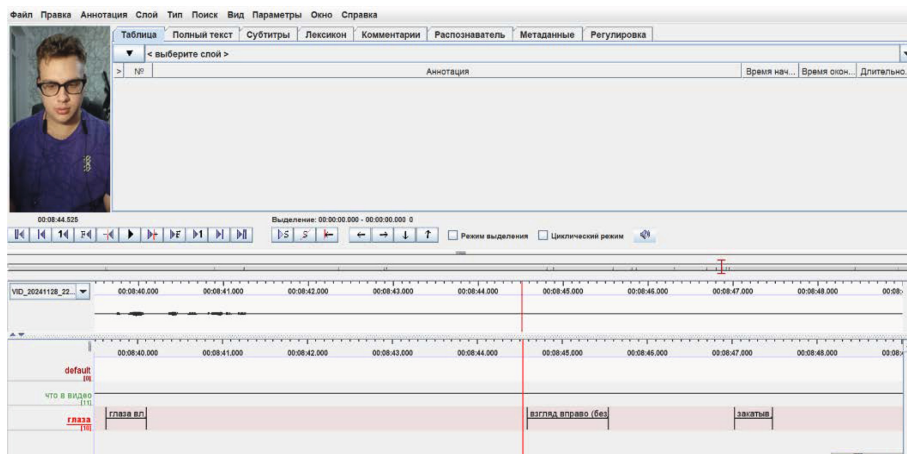


Fig. 3. Facial gesture annotation using ELAN.

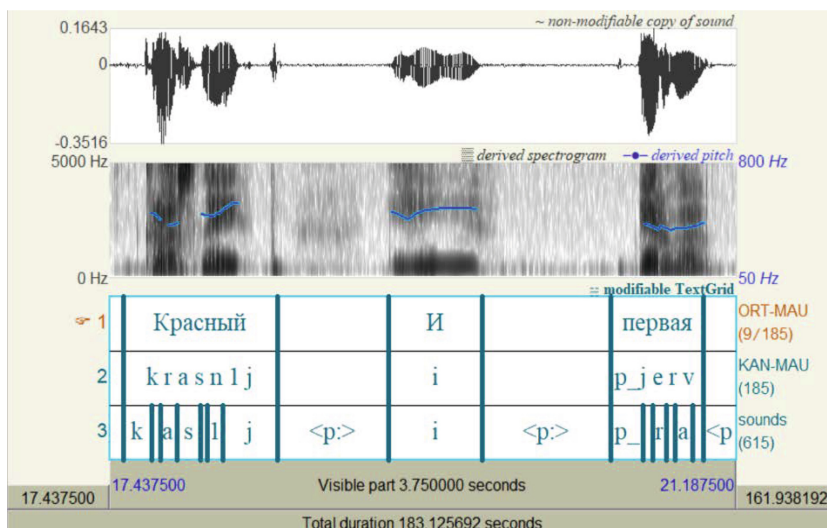


Fig. 4. Orthographic and phonetic annotation using Praat.

3.3 Results

Speech Parameters. The differences between sound duration with and without cognitive load are presented in Tables 5, 6 and 7. The sounds with the most noticeable differences in duration under the two conditions are listed in the following tables.

The differences in duration under the two conditions were the greatest in stressed vowels. The changes in the duration of consonants were smaller than those in the duration of vowels. Thus, cognitive load can probably be detected more effectively by the differences in the duration of vowels.

Table 5. The duration values of stressed vowels under cognitive load and without cognitive load.

	With cognitive load (ms)	Without cognitive load (ms)
/i/	179	52
/a/	162	90
/o/	118	72

Table 6. The duration values of unstressed vowels under cognitive load and without cognitive load.

	With cognitive load (ms)	Without cognitive load (ms)
/i/	121	37
/u/	91	56
/a/	101	69

Table 7. The duration values of sonorants and fricatives under cognitive load and without cognitive load.

	With cognitive load (ms)	Without cognitive load (ms)
/s/	124	70
/ʃ/	130	82
/s'/	129	88
/l'/	98	55

The changes in stressed and unstressed vowels (except /a/ in post-nuclear segments) with and without cognitive load were statistically significant. The differences between the duration values were statistically significant for all the consonants listed in Table 7 except /ʃ/.

Action Unit Frequency. The findings based on manual annotation showed that cognitive difficulties when answering the questions were most frequently marked by averting eyes up to the right (AU62 + AU63, according to the FACS) or down to the right (AU61 + AU63). In the majority of cases, averting eyes to the right was accompanied by turning head in the same direction. The frequency values of action units found by means of OpenFace are outlined in Table 8.

As shown in Table 8, cognitive load is generally accompanied by movements in the eye area and in the lip area. Dimpler, that draws lip corners back to the teeth, demonstrated the highest frequency. Regarding AU25 (lips part), this action unit might not be a reliable indicator of cognitive load as it might be attributed to articulation processes.

Table 8. The seven most frequent action units (recognized automatically).

Action unit number	Action unit name	Average frequency, %
AU14	Dimpler	66
AU05	Upper lid raiser	41
AU04	Brow lowerer	39
AU10	Upper lip raiser	37
AU45	Blink	36
AU25	Lips part	31
AU02	Outer brow raiser	24

Action Unit Intensity. The intensity values of action units obtained automatically are presented in Table 9.

Table 9. The seven action units with relatively high intensity values (recognized automatically).

Action unit number	Action unit name	Average intensity
AU07	Lid tightener	0.912
AU14	Dimpler	0.846
AU10	Upper lip raiser	0.659
AU04	Brow lowerer	0.621
AU25	Lips part	0.566
AU17	Chin raiser	0.512
AU26	Jaw drop	0.464

As shown in Table 8 and Table 9, certain facial movements (namely, AU14, AU10, AU04 and AU25) show relatively high values in both frequency and intensity. Dimpler (AU14) was in the second place in frequency and remained in the second place in intensity. Upper lip raiser (AU10) was the fourth most frequent action unit. Regarding intensity, it maintained its relative position among other action units to some extent since it is in the third place in intensity. A similar situation is observed for brow lowerer (AU04). On the contrary, upper lid raiser (AU05) was considerably frequent but demonstrated a relatively low intensity. In addition, the most intensive action unit, AU07 (lid tightener) did not show a high frequency value. Therefore, action unit intensity does not always correlate with frequency.

Both tables suggest that cognitive load can be detected on the basis of facial gestures in the eye area and in the lip area. Dimpler, upper lip raiser and brow lowerer appear to be the most significant characteristics due to their high frequency and intensity.

4 The Analysis of a Driver’s Speech and Mimics in the Wild

4.1 Materials

To study speech when driving and simultaneously talking in real-life conditions, three episodes of a talk show were analyzed. In each episode, the presenter is driving a car and interviewing the guest who is in the front seat next to him. Some parts of the interviews are set in the studio environments.

4.2 Methods

Firstly, all the fragments set in the car were separated from the scenes shot in the studio. Secondly, the scenes in which the driver is speaking were extracted from those fragments (FullHD 1920×1080 screen resolution, mp4 format). The total duration of the materials was 2 h 42 min.

The resulting videos were processed using OpenFace 2.2.0. As in the driving simulator experiment, action units were extracted from the video recordings. The driver’s facial expressions when driving and speaking at the same time were compared to his facial expressions in all the scenes showing him driving (regardless of whether he is speaking simultaneously or not). His speech characteristics when holding an interview in the car and in the studio were compared (Fig. 5).

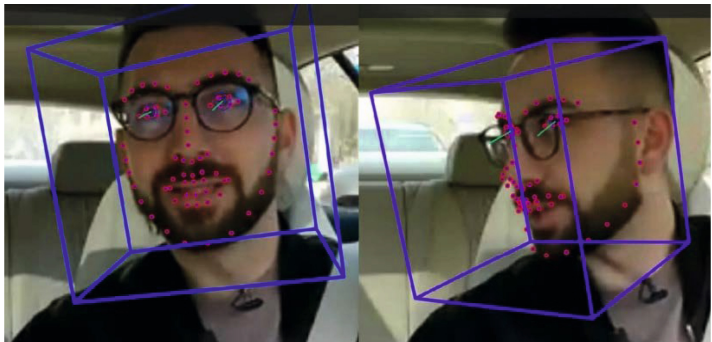


Fig. 5. The visualization of facial landmarks, eye gaze and head pose of a driver.

Sound tracks were extracted from the videos. The sound files were transcribed automatically using WEBMAUS Basic. The resulting orthographic and phonetic transcriptions were corrected manually using Praat. Acoustic parameters were obtained using SpeCT (The Speech Corpus Toolkit for Praat) scripts. The duration values were rounded to the nearest whole number.

4.3 Results

Speech Parameters. The results regarding duration values in the driver’s speech are presented in Tables 10, 11 and 12. The sounds with the most noticeable differences in duration under the two conditions are listed in the following tables.

Tables 10, 11 and 12 show that the duration of vowels and consonants increased when performing driving and speaking simultaneously. The mean differences between sound durations under the two conditions were generally higher than the results of the Stroop test experiment indicated. When performing a dual task, the duration of stressed and unstressed vowels was higher by 177 and 141 ms respectively compared to the interviewer's speech in the studio. The difference between the duration values of sonorants under the two conditions was, on average, 132 ms. Regarding fricatives, only one sound of this type (/z/) showed lengthening in the presence of high cognitive load.

Table 10. The duration values of stressed vowels under cognitive load and without cognitive load.

	Driving and speaking (ms)	Speaking (ms)
/a/	214	85
/o/	302	82
/i/	236	54

Table 11. The duration values of unstressed vowels under cognitive load and without cognitive load.

	Driving and speaking (ms)	Speaking (ms)
/a/	164	43
/i/	177	53
/i/	233	54

Table 12. The duration values of sonorants and fricatives under cognitive load and without cognitive load.

	Driving and speaking (ms)	Speaking (ms)
/m/	187	56
/n/	174	42
/z/	213	74

Unlike the findings of the Stroop test experiment, these results were statistically significant in most cases. The differences between stressed vowels /o/ and /i/ when speaking in the studio and speaking while driving showed high statistical significance. The differences between unstressed vowels /i/ (in post-nuclear parts of phonetic words) and /a/ (in pre-nuclear and post-nuclear segments) were also statistically significant,

likewise the differences in the duration of the sonorant /m/. However, the results regarding the sounds /n/ and /z/ were not statistically significant.

Action Unit Frequency. The frequency values of action units obtained from all the scenes that show the interviewer driving regardless of whether he is speaking simultaneously or not are presented in Table 13. The frequency values for action units when driving and speaking simultaneously can be seen in Table 14.

As shown in Table 13 and Table 14, the most frequent action unit in both situations was AU14 (71% when driving overall and 67% when driving and speaking simultaneously). Other action units with relatively high frequency under both these conditions were AU01, AU02, AU05, AU10, AU15 which are movements in the eye area. There seem to be no significant differences in the relative frequencies of different action units in Table 13 and Table 14.

Table 13. The action unit frequency values averaged for all the driving scenes.

Action unit number	Action unit name	Average frequency, %
AU14	Dimpler	71
AU05	Upper lid raiser	54
AU15	Lip corner depressor	43
AU01	Inner brow raiser	40
AU10	Upper lip raiser	39
AU02	Outer brow raiser	38

Table 14. The action unit frequency values averaged for the fragments with simultaneous driving and speaking.

Action unit number	Action unit name	Average frequency, %
AU14	Dimpler	67
AU02	Outer brow raiser	47
AU05	Upper lid raiser	43
AU01	Inner brow raiser	42
AU10	Upper lip raiser	41
AU15	Lip corner depressor	37

Dimpler (AU14) also showed the highest frequency in the simulator experiment. The frequency of this action unit in the first experiment was similar to its frequency for speaking in real-life driving conditions (66% and 67% respectively). Nevertheless, some differences are observed between the results of the two experiments. Regarding

the talk show fragments in which the interviewer was driving and speaking, AU01 and AU02 (inner and outer brow raising) demonstrated relatively high frequency values of 42% and 47% respectively. These values are approximately twice as high as the frequencies of these action units in the simulator experiment, which were 21% and 24%. Since driving in real conditions probably induces higher cognitive load than playing a simulator game, this difference may indicate that brow raising can serve as an indicator of a more significant cognitive load.

Action Unit Intensity. The results regarding the intensity of action units in all the driving scenes are shown in Table 15.

Table 15. The action units with the highest average intensity values (averaged over all the fragments set in the car).

Action unit number	Action unit name	Average intensity
AU14	Dimpler	1.401086
AU10	Upper lip raiser	1.027073
AU25	Lips part	0.840993
AU17	Chin raiser	0.799847
AU26	Jaw drop	0.706391
AU04	Brow lowerer	0.680125

The intensity values of the action units observed while driving and speaking simultaneously are listed in Table 16.

Table 16. The action units with the highest average intensity values (averaged over the fragments in which the driver is operating the vehicle and speaking).

Action unit number	Action unit name	Average intensity
AU14	Dimpler	1.167815
AU10	Upper lip raiser	1.014202
AU17	Chin raiser	0.611325
AU25	Lips part	0.541910
AU04	Brow lowerer	0.529352
AU26	Jaw drop	0.488872

As shown in Table 15 and Table 16, the set of the most intensive action units was the same for driving overall and for speaking while driving. For both conditions, these action units were positioned in relation to each other in ascending order nearly in the same way. The only difference was in the order of AU04 and AU26. In Table 15, AU26

preceded AU04, whereas AU04 preceded AU26 in Table 16. When a dual task (speaking) was performed, the action unit intensity was generally lower compared to all the driving scenes in total. This might suggest that facial expressions become less noticeable at higher levels of cognitive load.

5 Conclusion

Phonetic and visual characteristics of cognitive load were studied. The findings showed that under cognitive load the durations of both vowels and consonants increase while F0 range becomes narrower. Laryngealized segments also demonstrate higher duration under cognitive load. When speaking and driving simultaneously, which induces higher cognitive load, sound durations increased to a greater extent compared to the Stroop test experiment. It might suggest a direct correlation between sound duration and cognitive load level.

Regarding visual parameters, it was revealed that cognitive load can be recognized by muscle movements in the eye area and in the lip area. Dimpler, upper lip raiser and brow lowerer demonstrated relatively high frequency and intensity values in both experiments. Dimpler was the highest and the most intensive characteristic in the majority of cases, which can mean that this action unit is one of the most reliable markers of cognitive load. However, the results of the two experiments differed from each other in some respects. Inner and outer brow raising were much more frequent in real-life driving conditions while performing a dual task compared to the driving simulator experiment. Therefore, these facial gestures could probably be typical of higher cognitive load levels.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.








References

1. Sweller, J.: Cognitive load during problem solving: effects on learning. *Cogn. Sci.* **12**(2), 257–285 (1988)
2. Paas, F.G.W.C., van Merriënboer, J.J.: Instructional control of cognitive load in the training of complex cognitive tasks. *Educ. Psychol. Rev.* **6**(4), 351–371 (1994)
3. Baddeley, A.D.: Working memory. *Curr. Biol.* **20**(4), 136–140 (2010)
4. Le, P.N.: The Use of Spectral Information in the Development of Novel Techniques for Speech-Based Cognitive Load Classification: PhD thesis. Sydney (2012)
5. Yap, T.F.: Speech Production Under Cognitive Load: Effects and Classification. PhD thesis. Sydney (2012)
6. Huttunen, K., Keränen, H., Väyrynen, E., Pääkkönen, R., Leino, T.: Effect of cognitive load on speech prosody in aviation: evidence from military simulator flights. *Appl. Ergon.* **42**(2), 348–357 (2011)
7. Berthold, A., Jameson, A.: Interpreting symptoms of cognitive load in speech input. In: *Proceedings of the Seventh International Conference*, pp. 235–244. Springer, Vienna (1999)
8. Müller, C., Großmann-Hutter, B., Jameson, A., Rummer, R., Wittig, F.: Recognizing time pressure and cognitive load on the basis of speech: an experimental study. In: Bauer, M., Gmytrasiewicz, P.J., Vassileva, J. (eds.) *International Conference on User Modeling*, pp. 24–33. Springer-Verlag, Heidelberg (2001)

9. Khavylo, A., Engalychev, V., Leonova, E., Danshin, V., Mirzeabasov, O.: Manifestation of task's cognitive complexity in mimic micromovements: prognostic model. In: Arai, K. (eds.) *Proceedings of the Future Technologies Conference (FTC) 2021*, vol. 2. FTC 2021. *Lecture Notes in Networks and Systems*, vol. 359, pp. 256–267. Springer, Cham (2021)
10. Leonova, E., Engalychev, V., Khavylo, A., Mirzeabasov, O., Danshin, V.: Mimic indicators of task complexity: individual approach. In: *14th Conference of the European Human Behavior and Evolution Association*, pp. 199–200 (2019)
11. Ekman, P., Friesen, W.V.: *Manual for the Facial Action Coding System*. Consulting Psychologists Press, Palo Alto (1978)
12. OpenFace 2.2.0. <https://github.com/TadasBaltrusaitis/OpenFace/wiki>. Accessed 01 June 2025
13. Ivanko, D., Ryumin D., Axyonov, A., Kashevnik, A., Karpov, A.: Multi-speaker audio-visual corpus RUSAVIC: Russian audio-visual speech in cars. In: *Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022)*, pp. 1555–1559. European Language Resources Association (ELRA), Marseille (2022)
14. Stroop, J.R.: Studies of interference in serial verbal reactions. *J. Exp. Psychol.* **18**(6), 643–662 (1953)
15. Scarpina, F., Tagini, S.: The stroop color and word test. *Front. Psychol.* **8**, 557 (2018)
16. City Car Driving. <https://citycardriving.ru>. Accessed 06 Aug 2025



Cognitive Humor Processing in the Russian and English Internet Meme Chatting: EEG Study

Rodmonga Potapova¹ , Vsevolod Potapov^{1,2} , Ekaterina Karimova^{1,3} ,
Diana Smolskaya³ , Nikolay Bobrov¹ , Leonid Motovskikh¹ ,
and Iurii Pozhilov⁴ 

¹ Moscow State Linguistic University, 38 Ostozhenka Street, 119034 Moscow, Russia
rkpotapova@yandex.ru

² Lomonosov Moscow State University, Leninskije Gory 1, 119991 Moscow, Russia

³ Institute of Higher Nervous Activity and Neurophysiology of RAS, 5A Butlerova Street,
117485 Moscow, Russia
smolskaia.dv18@physics.msu.ru

⁴ Moscow State University of Psychology and Education, 29 Sretenka Street, 127051 Moscow, Russia

Abstract. This study deals with the neural correlates of internet meme perception using EEGs with a focus on prefrontal brain regions associated with complex cognitive and emotional processing. Twenty-one adult subjects watched memes and control stimuli with congruent and incongruent text-image pairs presented sequentially to isolate the periods of cognitive insight. Wavelet analysis revealed a specific pattern of brain activity around the fifth second of stimulus presentation—approximately one second after the image onset—within the alpha (11–14 Hz) and beta (22–28 Hz) frequency bands. This activity was localized in both the orbitofrontal and dorsolateral prefrontal cortices and was significantly noticeably more manifested for memes than for control stimuli. These results suggest that memes trigger neural responses associated with insight and semantic reinterpretation. Additionally, a language-dependent effect was observed: memes in Russian evoked greater dorsolateral prefrontal activity, while English memes entailed stronger orbitofrontal activation. These findings indicate that meme comprehension activates distinct neural mechanisms depending on the linguistic and cultural context of the viewer.

Keywords: Internet Memes · EEG · Wavelet Transform · Insight · Humor Processing · Prefrontal Cortex · Orbitofrontal Cortex · Dorsolateral Prefrontal Cortex · Language · Cognitive Neuroscience

1 Introduction

Memes constitute a phenomenon that has gained significant influence over the manner of communication and mode of thinking in the modern world, and, what matters even more, it affects patterns of global international communication and, eventually, decision

making, which leads to a conclusion that the importance of understanding the underlying nature of memes is exceptionally high and underestimated. Therefore memes should be regarded (and are treated in this paper) as the subject of a comprehensive scientific study the outcome of which will eventually become necessary for the prevention of dire consequences that may result from the global collective subconscious mind spinning out of control. While pure linguistics, mathematics or information science sometimes can afford being merely descriptive and more a matter of art rather than science, understanding of the phenomenon of memes can be deemed as important as that of the dangers that have emerged lately from the over-development of artificial intelligence. This is especially so because of the convergent and mutually fostering nature of the two, with the memes often being generated with the help or even by artificial intelligence, and the memes integrating human intelligence into artificial intelligence and not vice versa. Images and ideas in the generated memes are suggestive, which may mean, among all other aspects, that the way of thinking of the human recipients of memes may well be guided (or misguided) by that of artificial intelligence. This issue will probably become a prominent theme in our future studies. In our present-day research we concentrate on the phenomenon of the meme itself, on the insight in its inner neurophysiological and cognitive mechanisms that are the cause of its vast influence.

In the meantime, there is not even a consistent definition of what the meme is, leave alone understanding the reasons of the effects it causes. What can be clearly observed, however, is that these effects are at least noticed. “What’s in a meme? The Internet proffers almost an infinity of answers, mostly by way of examples, which go on to teasingly court and taskingly contort definition in myriad ways, leaving meming an enigmatic signifier—and memes sublime objects—to say the least ... Conceptually born into this world as an eminently adaptable element, it has to be remembered that this entails not only being adaptable to new conditions, but adaptable by them: the Internet (with its ads and apps) has, transformationally and irrevocably, adapted the meme” [3, 17, 18].

Now that the meme became a subject of scientific studies, these studies had to be assigned a domain. In our previous research we defined “memetics as an interdisciplinary field of knowledge, including, as an object of study, methods of transmitting network information with concise monocode or polycode (creolized) ministructures characterized by maximum network virality and popularity” [18, p. 80]. The term can also refer to the subject of this field, the memes in general in the context of scientific studies. From our point of view, the *memetics* completely coincides with the functions of **social network discourse (SND)** in an enlarged sense of the word. Mention should be made of the main distinctive features of SND, which are also significant for understanding the specific nature of memetics in digital communication. Identification of the verbal and paraverbal aspects of the formation and functioning of the SND in the global electronic media environment is based on its definition as a special electronic macropolylogue, taking into account the following types of categories of form, content and functional

weight[17]: a) electronic macropolylogue of a special SND form: distant; mediated; real-time (online) and delayed (offline); single-vector - multi-vector; monochronic - polychronic; b) electronic macropolylog with special SND content: monotopical - polytopical; information-rich (high context) - not information-rich (low context); provoking controversy, specific actions and deeds - not provoking controversy, specific actions and deeds; c) electronic macropolylogue with a special SND function: informing, containing the message sender's point of view; influencing, containing special linguistic means of influencing the recipient of the message; encouraging, with a specific goal to commit specific actions and deeds (particularly destructive ones, this type is implemented according to the scheme "stimulus \rightarrow pragmatic reaction in the form of a specific destructive action"), manipulating the consciousness of the recipient; intended for a target limited group of users - for an unlimited number of users; d) electronic macropolylogue with SND that considers factors influencing the specifics of communication: psychological and physiological (for example, age, gender, pathological, emotional, etc.); ethnic; socio-economic; political and geopolitical; confessional; culturological; pragmatic; moral and ethical. Currently, memetics is analyzed mainly in connection with social and political topics, for example, in [2, 4, 10–14, 28] and others. In the above studies, polycode memes are the content dominant, including various manifestations of the real political life of a particular country. It should be emphasized that the data obtained showed the relevance of the meme as a pedagogical and sociological means of observation [26]. In memetics, there is a wide use of various codes that characterize SND types [16]. These can be images of chess pieces in the traditional style or images of animals, images of paintings and sculptures in the classical style. A special type of monocode and polycode memetics includes the image of a person belonging to a certain class indicative of his/her social status: celebrities from the artistic world, famous politicians (persons of public interest).

The most "visited" are the meme-segments of the Internet in English, German and Russian. The exchange of information (in chats) using memetics is the preferred means of SND, as it makes it possible to respond promptly to various events with respect to morals and choices of today's youth. The study we conducted on the material of three language segments of the Internet (Russian-language, German-language and English-American language) [18–22] demonstrated the huge information potential of communication in relation to the young users of these segments. Memetics-based SND made it possible for users to instantly respond to the most significant events in the world, as well as in their own language area. In this regard, we also studied that cognitive and neurophysiological re-coding of the processes in the brain reinforced by the constant and long-term use of the same foreign language stimuli-patterns, which leads to a change in the behavioral reactions of Internet users in the process of virtual network communication, as well as real communication [23]. Previously, we also collected a large-scale multimodal polycode linguistic database of memes using Big Data processing technologies and a deep annotation system for polycode texts [24].

In our previous pilot work [25] neurophysiological correlates of textual modulation of perception of visual stimuli using English and Russian-language memes and control stimuli were identified using instrumental neuroimaging methods. Memes constitute a very particular cross-cultural phenomenon, which is a combination of textual and

illustrative information, and their effect on the functional state of the brain appears to be little studied. By demonstrating the textual and illustrative part of the memes separately and registering the EEG, we discovered how the text modulates the subsequent perception of the drawing by activating the mechanisms of visual attention. Reading the text in the native language caused a greater response of theta activity in the associative sensory areas of the cortex, which is associated with a better understanding of the meaning of the text. At the same time, the perception of illustrations to English-language memes caused a greater response of theta and alpha rhythms in most of the considered areas of the cortex, which reflects the processes of memory, emotional reaction and the involvement of large neuronal resources for the integration and understanding of the whole image of the meme.

Given these preliminary results, we hypothesized that internet memes evoke distinct neural responses in the prefrontal cortex, reflecting processes of insight and recontextualization, which differ from those evoked by control stimuli with congruent or incongruent text-image pairs. We further expected language-dependent differences, with native and non-native meme processing engaging various prefrontal regions. The aims of the study were to identify the temporal and spectral characteristics of meme-related EEG activity, localize it within prefrontal areas, and compare neural responses to memes in Russian and English.

2 Method

2.1 Participants

A total of 21 individuals aged between 18 and 35 years participated in the study. None had a history of neurological or psychiatric ailments, hearing or vision impairments (corrected vision was allowed), traumatic brain injury within the prior three years, or were taking antidepressants at the time of the study. Additionally, participants completed a baseline English proficiency test and psychological screening questionnaires to exclude symptoms of depression and high anxiety (Beck Depression Inventory and Spielberger State-Trait Anxiety Inventory) on the day of the experiment. Only subjects with B1–B2 level of English proficiency and no signs of depression according to the Beck scale were included in the sample. The study was conducted at the Center for Cognitive Psychophysiology, Institute of Higher Nervous Activity and Neurophysiology, Russian Academy of Sciences (IHNA & NPh RAS), and involved healthy adult volunteers. The study protocol was approved by the Ethics Committee of the IHNA & NPh RAS.

2.2 Stimuli

The first stage of the study involved choosing suitable stimuli. Internet memes of a particular format were sourced online: a piece of text that presented a new context and altered the interpretation of an accompanying image. A total of 215 meme-format images served as stimuli (top text in a framed box, image beneath), divided into six categories:

Memes:

- 45 Russian-language memes (text + image) – ***Memes RU***;
- 35 English-language memes (text + image) – ***Memes EN***;

Control Stimuli:

- *Description matches*:
 - 40 Russian-language congruent control stimuli (image with contextually matching description) – ***Description matches RU***;
 - 35 English-language congruent control stimuli – ***Description matches EN***;
- *Description mismatches*:
 - 30 Russian-language incongruent control stimuli (image with unrelated description) – ***Description mismatches RU***;
 - 30 English-language incongruent control stimuli – ***Description mismatches EN***.

Memes were chosen to allow for separate, sequential presentation of the text and image on screen. This was necessary to temporally separate the perception of the text from the viewing of the image, enabling precise EEG synchronization with both events. Another condition was that the text alone had to be contextually incomplete; the image was required to complete the humorous intent. Neither the text nor the image per se should convey a full joke or be humorous in isolation. To ensure homogeneity of the stimulus set, images had to consist of a single photograph—comics or multi-frame illustrations were excluded. For the pilot experiment, 45 English-language and 35 Russian-language memes were chosen.

As control stimuli, photographs commonly used in memes were selected. Each image was paired with a contextually relevant but non-humorous description. Thus, 40 control stimuli with Russian descriptions and 35 with English descriptions (*Description matches RU/EN*) were created. Another category of control stimuli consisted of images paired with unrelated descriptions in Russian and English (*Description mismatches RU/EN*). Since memes typically contain a non-literal relation between image and text (often forming an allegory), these incongruent control pairs helped to distinguish memes from entirely mismatched content. Text and image were presented sequentially to capture the brain's response to each component separately. First, the text was shown for 4 s while the image remained blurred, followed by the image for 4 s with the text blurred (see Fig. 1).



Fig. 1. Experimental paradigm: sequence of stimulus presentation and example stimuli for each category.

2.3 Experimental Procedure

First, the subjects signed informed consent and data processing agreements, then completed the preliminary English proficiency test. The experiment was conducted in a room isolated from external stimuli. Stimulus presentation and experimental design were implemented using Presentation software (Neurobehavioral Systems Inc.). A 64-channel EEG cap (actiCHamp, Brain Products GmbH, Germany) was applied. Calibration trials were recorded with eyes open and closed, followed by pseudo-randomized visual stimulus presentation. First, the meme’s textual component was presented on a gray background for 4 s, then the corresponding image for another 4 s was shown. Each stimulus pair was separated by a 3–5 s inter-stimulus interval. In the control conditions,

the context description appeared first for 4 s, followed by the matching or mismatching image for 4 s (Fig. 1).

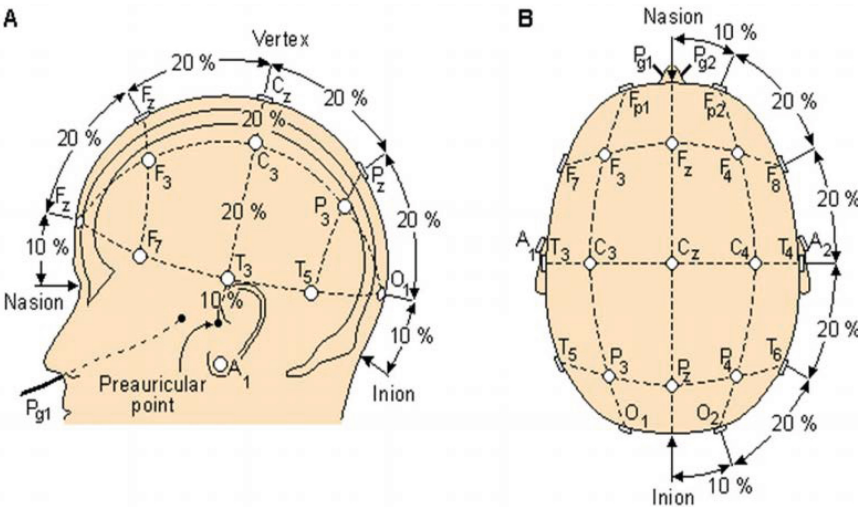


Fig. 2. International electrode placement system.

2.4 EEG Recording

A 64-channel EEG was recorded using a DC amplifier (BrainAmp by Brain Products GmbH, Germany) and Ag/AgCl electrodes arranged according to the international 10–20 system, with FCz as the reference (Fig. 2). Data were sampled at 512 Hz; the impedance level was maintained below 15 kΩ. A low-pass filter at 70 Hz, high-pass filter at 1 Hz, and notch filter at 50 Hz were applied. The recorded EEG data were analyzed with the MNE-Python software package. Preprocessing steps included band-pass filtering (1–35 Hz) and artifact removal using Independent Component Analysis (ICA). Independent components were computed using the “infomax” algorithm. Topographic maps were generated from the demixing matrix coefficients to identify the spatial distribution of each component across the scalp model.

2.5 Wavelet Analysis of EEG

Following preprocessing, time-frequency analysis was performed with the Morlet wavelet transform. Signal epochs were segmented according to stimulus onset markers. A 1-s baseline interval immediately preceding the stimulus onset was used. Wavelet maps were averaged individually for each electrode and stimulus category. The resulting three-dimensional wavelet maps had frequency bands along the y-axis, covering four standard EEG rhythms: theta (4–7 Hz), alpha (8–13 Hz), lower beta (14–24 Hz), and upper beta (25–35 Hz). The x-axis represented time from the stimulus onset (0 s) to the end of the image presentation (8 s). The transition from text to image occurred at the 4-s mark. Wavelet data were then averaged across two cortical regions: the orbitofrontal

cortex and the dorsolateral prefrontal cortex, yielding a set of 3D wavelet maps for each stimulus category in both regions.

2.6 Statistical Analysis

Statistical analyses were performed using MNE-Python and Statistica (StatSoft). The main approach used to compare wavelet maps generated for stimulus categories was a non-parametric cluster-based permutation test for paired samples. This method involves combining data from different conditions into a single dataset, followed by random reassignment of trials into new subsets and the calculation of the test statistic for each permutation. Multiple reiteration of this process allows for the construction of a reference distribution, against which the observed test statistic is compared. Clusters with $p < 0.05$ were considered statistically significant.

3 Results

3.1 Comparison of the Three Stimulus Categories Regardless of Language

The permutation test indicated notable differences when analyzing the “memes” category in comparison to the two control stimulus categories at the fifth second of stimulus presentation. These differences were identified in the 11–14 Hz and 22–28 Hz frequency bands (the second harmonic), in both the orbitofrontal and dorsolateral prefrontal cortices (see Fig. 3; the significant cluster is marked with a yellow frame).

3.2 Comparison of the “Meme” Category in English and Russian

In the subsequent phase, we analyzed the cluster identified previously, which focused on meme perception and not observed for control stimuli. We compared its activation across the two language conditions (Russian and English) in both the orbitofrontal and dorsolateral prefrontal cortices. At the fifth second of stimulus presentation, meme perception in Russian triggered significantly greater amplitude activity in the dorsolateral prefrontal cortex, whereas meme perception in English showed significantly greater amplitude activity in the orbitofrontal cortex (see Fig. 4).

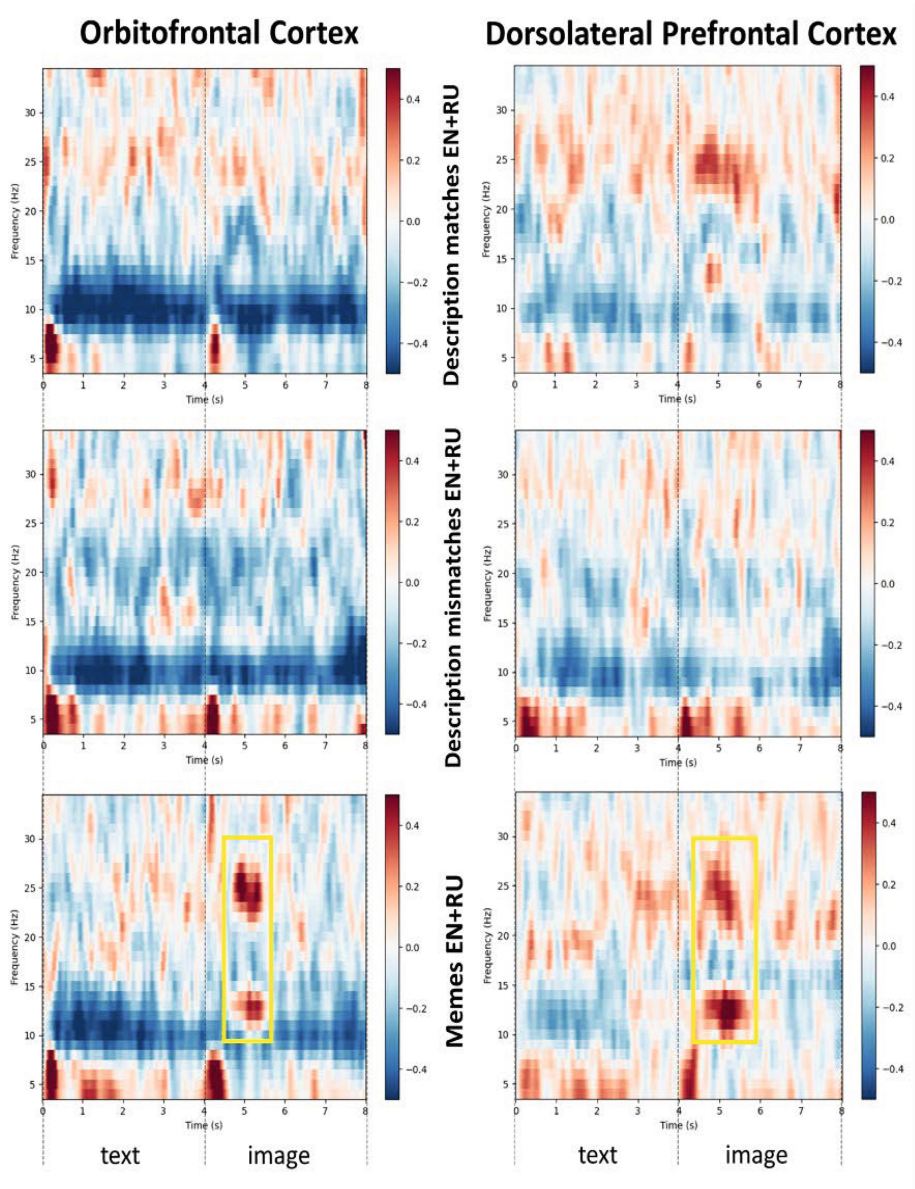


Fig. 3. Three-dimensional wavelet maps illustrating the dynamics of brain activity in the orbitofrontal and dorsolateral prefrontal cortices during the perception of the textual (1–4 s) and visual (4–8 s) components of the stimulus. Averaged maps are shown for the three stimulus categories (averaged across languages): control stimuli with congruent descriptions, incongruent descriptions, and memes. Clusters that differ noticeably from the control stimuli during meme perception are shown in yellow frames.

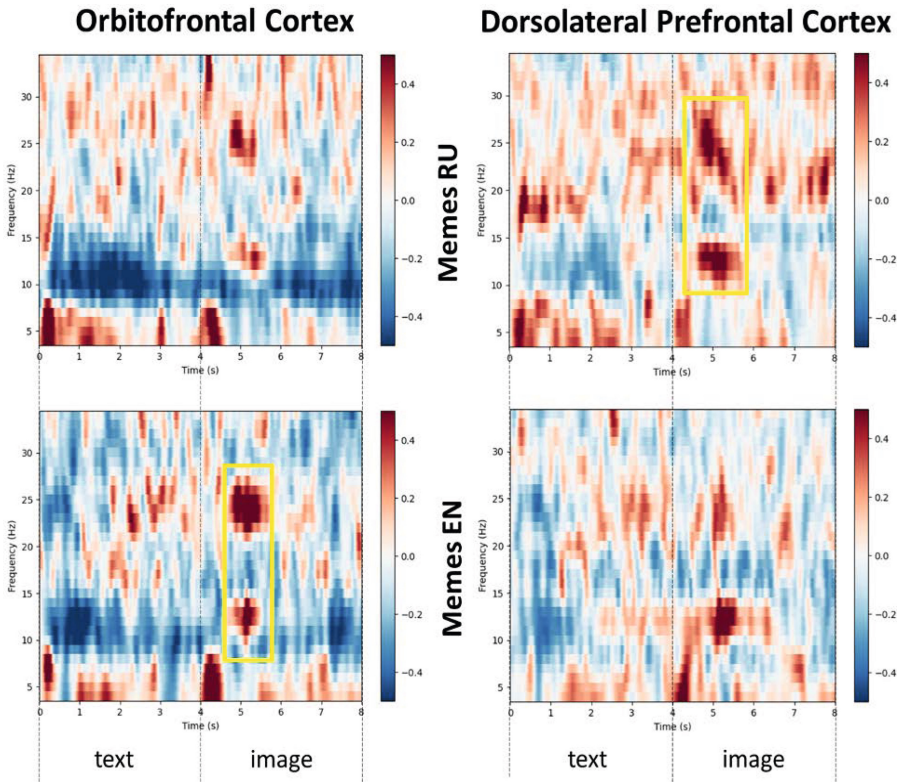


Fig. 4. Three-dimensional wavelet maps illustrating the dynamics of brain activity in the orbitofrontal and dorsolateral prefrontal cortices during the perception of the textual (1–4 s) and visual (4–8 s) components of the stimulus. Averaged maps are shown for memes in Russian (top panels) and English (bottom panels). Regions showing greater amplitude activity for each language category, within the previously identified cluster, are shown in yellow frames.

4 Discussion

The aim of the present study was to identify distinct patterns of brain activity linked to the perception of internet memes, as opposed to control stimuli, within two regions of the prefrontal cortex involved in higher-order cognitive processing. The control stimuli – including either congruent or incongruent descriptions – were introduced to distinguish the general effect of text-image incongruity from the more complex mechanism of insight-based humor, where the text generates a new interpretative context for the image. To pinpoint the temporal dynamics of insight formation, we separated the presentation of textual and visual components. We hypothesized that insight – the recognition of the hidden, often humorous meaning – would occur approximately 0.5 s after the onset of the visual stimulus. Our findings supported this assumption: specific neural activity patterns emerged around the fifth second of stimulus presentation, which corresponded to about one second after the image was revealed. These patterns were notably absent in control conditions.

This observed activity was localized in both the orbitofrontal cortex (OFC) and the dorsolateral prefrontal cortex (DLPFC). These findings align with previous studies analyzing these regions in the processing of humor, insight, and emotional evaluation. The OFC has been associated with reward processing and affective valuation [8], including the appreciation of humor [29], while the DLPFC plays a central role in cognitive control, integration of abstract information, and restructuring of mental representations—all critical for insight-driven problem solving [6, 9]. Interestingly, language-specific differences were observed in the distribution of this insight-related activation. Russian memes evoked greater DLPFC activity, whereas English memes evoked greater activation in the OFC. A potential explanation for this distinction is that processing a meme in one's native language (Russian, in this case) involves more extensive cognitive elaboration and conceptual integration - processes typically mediated by the DLPFC. In contrast, when processing memes in a second language (English), subjects may rely more on emotional and heuristic processing, which leads to increased involvement of the OFC, which is known for its role in rapid affective evaluations and integration of socially relevant cues [1, 27].

This dissociation may also reflect differential cognitive strategies depending on language proficiency. Non-native language processing has been shown to reduce emotional resonance and increase cognitive load [15]. As a result, the participants may rely more heavily on affectively salient cues in the image, processed predominantly by the OFC, instead of integrating text and image conceptually. Moreover, the observed oscillatory activity in the alpha and beta frequency bands aligns with existing evidence linking these bands to semantic integration, attentional engagement, and cognitive flexibility. Alpha desynchronization have been commonly linked to increased task engagement and semantic access [7], while beta-band increases have been linked to the maintenance of current cognitive sets and top-down control mechanisms [5]. The appearance of the second harmonic in the beta range may suggest increased demand for recontextualization and reinterpretation during meme comprehension.

Together, these findings support the notion that memes are not simply humorous images but rather complex cognitive stimuli that activate variation in neural activation in various language conditions also highlights the interaction between linguistic processing and higher-order cognition in the perception of culturally-nuanced visual humor.

5 Conclusion and Future Work

The results of the conducted research at this stage contain very valuable results, which in the future can be expanded, supplemented and more deeply interpreted from the position of experimental cognitive science of human speech behavior, the peculiarities of perception and interpretation of speech stimuli, containing a variety of structures of linguistic features of the speech signal, the degree of complexity of the linguistic material, etc. Thus, this stage of our research fully reflects the feasibility of further experiments, taking into account the degree of knowledge of a foreign language, the technique of instrumental analysis, the age and nationality of the subject, etc. In the future, it would be possible to also take into account the factor of the degree of emotionality of the subjects, etc. The development of the features of memetics that we have begun is advisable from

the point of view of various branches of science in general, starting with the patterns of communication, appeal to foreign languages, the degree of their knowledge, the emotions accompanying meme chatting, the features of recoding real speech from the native language into foreign languages, the emotional state of communicants, etc.

Acknowledgments. The research is supported by the Russian Science Foundation, grant №25–28-01595, scientific supervisor: Vsevolod Potapov, Dr. Sci.

References

1. Amodio, D.M., Frith, C.D.: Meeting of minds: the medial frontal cortex and social cognition. *Nat. Rev. Neurosci.* **7**(4), 268–277 (2006). <https://doi.org/10.1038/nrn1884>
2. Bown, A., Bristow, D. (eds.): *Post Memes: Seizing the Memes of Production*. Punctum books, Brooklyn (2019)
3. Bristow, D.: Introduction. In: Bown, A., Bristow, D. (eds.) *Post Memes: Seizing the Memes of Production*, pp. 17–24. Punctum books, Brooklyn (2019)
4. Egner, M.: *Humor im Internet Analyse humoristischer Formen der Kommunikation in sozialen Medien*. Masterarbeit. Universität Salzburg (2018). <https://eplus.uni-salzburg.at/obvusbhs/content/titleinfo/5015234/full.pdf>
5. Engel, A.K., Fries, P.: Beta-band oscillations—signalling the status quo? *Curr. Opin. Neurobiol.* **20**(2), 156–165 (2010). <https://doi.org/10.1016/j.conb.2010.02.015>
6. Jung-Beeman, M., et al.: Neural activity when people solve verbal problems with insight. *PLoS Biol.* **2**(4), e97 (2004). <https://doi.org/10.1371/journal.pbio.0020097>
7. Klimesch, W.: Alpha-band oscillations, attention, and controlled access to stored information. *Trends Cogn. Sci.* **16**(12), 606–617 (2012). <https://doi.org/10.1016/j.tics.2012.10.007>
8. Kringelbach, M.L.: The human orbitofrontal cortex: linking reward to hedonic experience. *Nat. Rev. Neurosci.* **6**(9), 691–702 (2005). <https://doi.org/10.1038/nrn1747>
9. Luo, J., Niki, K.: Function of the right dorsolateral prefrontal cortex in insight. *NeuroReport* **14**(5), 697–701 (2003). <https://doi.org/10.1097/01.wnr.0000059610.30203.69>
10. Milner, R.: *Pop Polivocality: Internet Memes, Public Participation, and the Occupy Wall Street Movement*. *International Journal of Communication* **7**, 2357–2390 (2013). URL: <https://ijoc.org/index.php/ijoc/article/view/1949/1015>
11. Milner, R.: *The world made meme: Discourse and identity in participatory media*. Kansas, University of Kansas, PhD diss (2012). <http://hdl.handle.net/1808/10256>
12. Nowotny, J., Reidy, J.: *Memes – Formen und Folgen eines Internetphänomens*. Bielefeld: Transcript-Verlag (2022). <https://www.transcript-verlag.de/media/pdf/70/97/c5/oa9783839461242.pdf>
13. Osterroth, A.: Das Internet-Meme als Sprache-Bild-Text. *IMAGE. Zeitschrift für interdisziplinäre Bildwissenschaft*, Heft 22, Jg. 11, Nr. 2, 26–46 (2015)
14. Osterroth, A.: *Sprache-Bild-Kommunikation in Imageboards – Das Internet-Meme als multimodaler Kommunikationsakt und politisches Ärgernis*. Universität Koblenz-Landau, Deutschland (2016)
15. Pavlenko, A.: Affective processing in bilingual speakers: Disembodied cognition? *Int. J. Psychol.* **47**(6), 405–428 (2012). <https://doi.org/10.1080/00207594.2012.743665>
16. Petz, A.: *Kategorisierung neuer Meme-Genre anhand einer Korpusanalyse. Lehre in den Digital Humanities. Ein Portal der IT-Gruppe Geisteswissenschaften der LMU* (2021). <https://www.dh-lehre.gwi.uni-muenchen.de/?abschlussarbeit=kategorisierung-neuer-meme-genre-anhand-einer-korpusanalys>

17. Potapova, R.: From deprivation to aggression: verbal and non-verbal social network communication. In: Global Science and Innovation. Materials of the VI International Scientific Conference, Chicago, USA, 18–19 November 2015, pp. 129–137 (2015)
18. Potapova, R., Potapov, V.: Internet memetics as an emotiogenic environment of the network communication. *Bull. Russ. Acad. Sci. Stud. Literature Lang.* **81**(2), 78–91 (2022a) (in Russian). <https://doi.org/10.31857/S160578800019458-9>
19. Potapova, R.K., Potapov, V.V.: Features of modern multilingual memetics. *Vestnik Moscow State Univ. Human.* **11**(866), 83–91 (2022) https://doi.org/10.52070/2542-2197_2022_11_866_83
20. Potapova, R.K., Potapov, V.V.: Internet memetics as an emotiogenic environment of the network communication. *Bull. Russ. Acad. Sci. Stud. Literat. Lang.* **81**(2), 78–91 (2022). (In Russian). <https://doi.org/10.31857/S160578800019458-9>
21. Potapova, R.K., Potapov, V.V.: Features of German digital memetics. *Vestnik Moscow State Linguist. Univ. Human.* **10**(878), 77–85 (2023). https://doi.org/10.52070/2542-2197_2023_10_878_77
22. Potapova, R.K., Potapov, V.V.: Memolect as a basis of meme chatting (on the material from three language Internet areas). In: Potapov, V.V., Kazak, E.A. (eds.) *New Regionalism: From Traditional Forms of Dialects to New Realities: Collective Monograph*, pp. 375–405. INION RAS, Moscow (2025), (in Russian)
23. Potapova, R., Potapov, V., Gorbunov, P. The brain activity of the bilingual code-switching communication. In: Wen, S., Yang, C. (eds.) *Biomedical and Computational Biology. BECB 2022. Lecture Notes in Computer Science*, vol. 13637, pp. 274–281. Springer, Cham (2023) https://doi.org/10.1007/978-3-031-25191-7_22
24. Potapova, R., Potapov, V., Gorbunov, P.: On the experience of statistical processing of memes in Big Data format. In: Yang, X.S., Sherratt, S., Dey, N., Joshi, A. (eds.) *Proceedings of Ninth International Congress on Information and Communication Technology. ICICT 2024. Lecture Notes in Networks and Systems*, vol. 1014, pp. 297–304. Springer, Singapore (2024) https://doi.org/10.1007/978-981-97-3562-4_24
25. Potapova, R., Potapov, V., Karimova, E., Motovskikh, L., Bobrov, N.: Neurophysiological Correlates of textual modulation in visual stimuli: an experimental study of Russian and English Memes. In: Karpov, A., DeliĆ, V. (eds.) *Speech and Computer. SPECOM 2024. Lecture Notes in Computer Science*, vol. 15299, pp. 201–215. Springer, Cham (2025) https://doi.org/10.1007/978-3-031-77961-9_15
26. Reidel, L.: *Eine konsumentenorientierte Betrachtung der Memetik*. GRIN Verlag, Munich (2019)
27. Rolls, E.T.: The functions of the orbitofrontal cortex. *Brain Cogn.* **55**(1), 11–29 (2004). [https://doi.org/10.1016/S0278-2626\(03\)00277-X](https://doi.org/10.1016/S0278-2626(03)00277-X)
28. Segev, E., Nissenbaum, A., Stolerio, N., Shifman, L.: Families and networks of internet memes: the relationship between cohesiveness, uniqueness, and quiddity concreteness. *J. Comput.-Mediat. Commun.* **20**(4), 417–433 (2015)
29. Vrticka, P., Black, J.M., Reiss, A.L.: The neural basis of humour processing. *Nat. Rev. Neurosci.* **14**(12), 860–868 (2013). <https://doi.org/10.1038/nrn3566>



Saudi Sign Language Translation Using T5

Ali Alhejab¹ , Tomáš Železný² , Lamya Alkanhal³ , Ivan Gruber² ,
Yazeed Alharbi¹ , Jakub Straka² , Václav Javorek² , Marek Hruží² ,
Badriah Alkalifah¹, and Ahmed Ali¹

¹ HUMAIN, Riyadh, Saudi Arabia
{aalhejab,yaharbi}@humain.ai

² Department of Cybernetics and New Technologies for the Information Society,
University of West Bohemia, Pilsen, Czech Republic
{zeleznyt,grubiv}@ntis.zcu.cz

³ Saudi Data & AI Authority, Riyadh, Saudi Arabia

Abstract. This paper explores the application of T5 models for Saudi Sign Language (SSL) translation using a novel dataset. The SSL dataset includes three challenging testing protocols, enabling comprehensive evaluation across different scenarios. Additionally, it captures unique SSL characteristics, such as face coverings, which pose challenges for sign recognition and translation. In our experiments, we investigate the impact of pre-training on American Sign Language (ASL) data by comparing T5 models pre-trained on the YouTubeASL dataset with models trained directly on the SSL dataset. Experimental results demonstrate that pre-training on YouTubeASL significantly improves models' performance (roughly 3× in BLEU-4), indicating cross-linguistic transferability in sign language models. Our findings highlight the benefits of leveraging large-scale ASL data to improve SSL translation and provide insights into the development of more effective sign language translation systems. Our code is publicly available at our GitHub repository (<https://github.com/signforall/t5-training-scripts>).

Keywords: Sign language translation · LLMs · T5 · Saudi sign language

1 Introduction

Sign languages (SLs) are rich, fully developed natural languages that serve as the primary means of communication for Deaf communities worldwide. Unlike spoken languages, SLs utilize visual-gestural modalities: hand shapes, movements, facial expressions, and body language, to convey meaning. According to recent estimates, more than 70 million Deaf people use SL, and there are over 300 distinct SLs in use globally, reflecting the diverse cultural and linguistic heritage of Deaf communities¹.

¹ <https://www.handtalk.me/en/blog/nteresting-facts-about-sign-languages/>.

In many countries, there is increasing recognition of the need for inclusive communication in public and private institutions such as banks, hospitals, and schools. Effective communication between the Deaf community and these institutions is essential for ensuring equitable access to critical services. For instance, in healthcare settings, the presence of qualified SL interpreters has been shown to significantly improve patient understanding and satisfaction, while in educational environments, SL is the preferred language for many Deaf students to learn complex concepts in their native tongue (see NAD Position Statement on Health Care Access for Deaf Patients, 2020²). Similarly, financial institutions and government agencies are progressively adopting SL interpretation services to better serve Deaf clients, highlighting the importance of culturally and linguistically appropriate communication.

One of the main challenges in sign language translation (SLT) is the scarcity of training data, particularly for sign languages that are underrepresented in publicly available resources. Recent research has explored SLT in a multilingual context by leveraging corpora from multiple sign languages (SLs), which not only helps address the data scarcity issue but also allows models to exploit shared linguistic structures, leading to improved translation quality [16]. This challenge is especially pronounced for under-resourced sign languages like Saudi Sign Language (SSL), primarily used by the Deaf community in Saudi Arabia. SSL is characterized by unique region-specific gestures, non-manual markers (such as facial expressions and body movements), and syntactic structures that reflect both cultural influences and elements of spoken Arabic. Unlike Unified Arabic Sign Language, a standardized system used across many Arab countries, SSL has developed independently, resulting in distinct grammar and vocabulary. For instance, [1] demonstrates that SSL follows unconventional sentence structures and word orders, differing from the broader Arabic SL standard, while [15] highlights unique non-manual markers and syntactic patterns that further differentiate SSL from other Arabic SL variants.

The main contributions of this paper are as follows: Firstly, we propose a processing pipeline directly tailored for sign language videos. Secondly, we demonstrate the effectiveness of pre-training on a different SL to improve generalization performance. We explore this idea by applying it to Saudi Sign Language (SSL), leveraging the ASL dataset YouTubeASL [17] to pre-train a T5-based SLT model. We then compare the results with a model trained from scratch. Using a pose-based approach that omits the appearance of signers, our findings show that cross-lingual pre-training significantly enhances performance, highlighting its potential for low-resource SLs.

2 Related Work

Sign Language Translation has advanced through both gloss-based approaches [2, 4, 23], which use glosses - structured linguistic representations of signs - for improved alignment, and gloss-free approaches [8, 21, 22], which aim

² <https://shorturl.at/tQ1De>.

Table 1. Dataset splits with details on number of sentences (Sents), minutes (Min), seen sentences/signers, etc.

Split	Sents	Min	Seen Sents	Seen Signers	# Samples	# Signers	Gender
Train	24,111	2,017.82	✓	✓	1,900	16	4F, 12M
Test 1	200	16.65	x	x	100	2	1F, 1M
Test 2	1,297	107.95	x	✓	100	11	3F, 10M
Test 3	3,783	337.33	✓	x	1,900	2	1F, 1M

to learn direct mappings from visual features to text. While gloss-based methods benefit from explicit supervision, recent gloss-free approaches have become increasingly popular by utilizing multimodal learning techniques. The advancement of Large Language Models (LLMs) has further improved gloss-free SLT, as seen in [11, 14, 18], by the use of better pre-trained textual representations to improve translation accuracy.

While studies about SSL like have focused on recognition rather than full translation, multilingual corpora such as [7] have shown the potential for cross-lingual adaptation. Bilingual transfer methods, like the ones used in [10], show a high added value when using high-resource SLs to improve translations for lower-resource ones.

Several large-scale datasets are being used for SLT training [3, 5, 16, 17]. However, privacy concerns and high annotation costs limit their scalability. In [14], the authors take this issue on by introducing self-supervised pre-training on anonymized videos, and [11] uses hierarchical visual encoders and multimodal tuning to find better sign language representations without gloss supervision.

Transformer architectures, such as the Text-to-Text Transfer Transformer T5 [13], have demonstrated significant effectiveness in SLT due to their encoder-decoder structure and multilingual capabilities. Studies [6, 20] have demonstrated T5’s adaptability to multimodal input. Our work builds on this, using T5 as an SLT baseline while addressing the data limitations of SSL by employing multilingual transfer learning.

3 Data

The dataset used in this study is the Saudi Sign Language corpus, which belongs to under-resourced SLs. In comparison with datasets such as YouTubeASL [17], the number of recorded hours is significantly lower. However, one key advantage of SSL is that it enables a thorough assessment of model generalization.

3.1 Dataset Composition

The SSL dataset comprises 2,000 unique sentences, representing common expressions in the deaf community, spanning everyday communication and specialized

domains (banking, law, education, healthcare, emergency services, and transportation). The original sentences are in Arabic and were translated into English for our experiments, as T5 and T5 v1.1 only support English. For mT5, which is multilingual, both the original Arabic sentences and the translated English sentences were used. The temporal distribution of the data is illustrated in Fig. 1, which highlights variations in sentence length and signing duration.

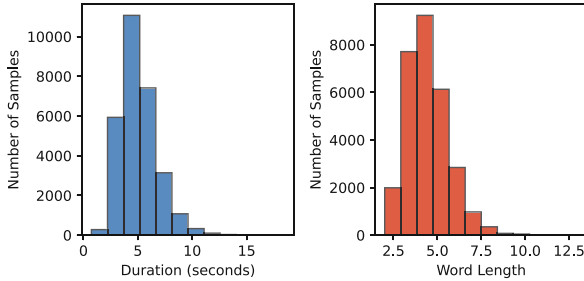


Fig. 1. Histogram of Signing Duration and Word Length.

Eighteen signers participated in the recording, with 13 males and 5 females, resulting in a higher male representation. Notably, female signers’ faces were intentionally obscured (e.g., with masks or veils), while male signers’ faces remained visible.

3.2 Data Splits

The dataset is divided into train and multiple test splits (Table 1). Each test scenario is designed to assess the model’s ability to handle varying degrees of data exposure and distribution shifts. The test scenarios are as follows:

Test 1: This test is designed to assess the model’s generalization ability to entirely unseen data. While the unseen sentences are unseen as a whole, they are composed of words seen during training. It evaluates the model’s performance on both unseen sentences and signers, providing a measure of its robustness to novel input during inference.

Test 2: In this scenario, the model is tasked with generating translations for unseen sentences, but with signers it has encountered during training. This evaluates the model’s ability to generalize to new, out-of-distribution sentences while leveraging prior knowledge of familiar signers and their signing styles.

Test 3: This test examines the model’s performance on sentences it has been exposed to during training, but they are performed by unseen signers. This serves to gauge how well the model can generalize to the variety of ways people sign the same words.

The split highlights the key advantages of the dataset, as it provides more insight into model performance and generalization. Contrary to prior works, the splits enable direct measurement of whether the model is sensitive to signer appearance (Test 1 and Test 3) and word order (Test 1 and Test 2).

3.3 Key Challenges and Limitations

Besides the challenging test splits, the SSL dataset presents a few challenges; some of them specific to SSL, and some of them are more general. The challenges in the data are as follows:

- **Face Occlusion:** The covering of female faces may limit the model’s ability to learn important features from the lips or signs that rely on facial expression, such as question marks.
- **Gender Imbalance:** The dataset includes SL data from 18 signers, with a notable gender imbalance (more male signers than female)
- **Unbalanced Data Across Domains:** While the dataset spans multiple domains, the distribution of sentences across these domains may not be uniform.

4 Methods

4.1 Video Preprocessing

SL datasets vary in recording conditions. Some, like How2Sign [5], are captured in controlled environments with a single signer centered in the frame. Others, such as YouTubeASL [17], contain videos recorded in the wild, where signers may appear at different distances from the camera, in varying positions, or alongside multiple people.

To standardize the data, we first preprocess the videos to ensure that the signing individuals are centered in the frame, have a normalized size, and that all videos have the same resolution across the dataset. Additionally, we extract pose features during this step.

Our preprocessing pipeline consists of multiple steps. First, we use lightweight YOLOv8-nano [9] to detect the rough body pose of all individuals in the frame. To simplify processing, we discard videos with multiple people, as tracking multiple individuals and identifying the signer throughout the video introduces complexity and potential misalignment between signing and translations. This step is utilized for the YouTubeASL dataset.

Next, we define the signing space, which in SL linguistics refers to the area where signing occurs. Inspired by [2], we represent this as a box centered between the shoulders, with a height and width four times the shoulder distance. If body pose keypoints fall outside this box, we expand it to include them. To create a stable bounding box for the entire video, we compute the signing space for each frame and take the median of the coordinates; this mitigates fluctuations caused by detection errors.

We then refine the signer’s pose using MediaPipe [12], a more precise model for body, hand, and facial keypoints. MediaPipe performs better when the signer is centered in the frame, which especially benefits face and hand detection. Using the updated body keypoints, we adjust the signing space. In some cases, it is also necessary to determine the handedness of detected hand keypoints based on

the Euclidean distance between wrist keypoints from the body pose and hand pose predictions.

For each hand, we obtain 21 keypoints. For body pose, we start with 33 keypoints but remove those corresponding to legs, as they are not essential for translation, leaving 25 keypoints. Lastly, we extract a dense face mesh containing 478 keypoints, from which we select 37 keypoints³ that represent facial features. In total, we extract 104 keypoints. Figure 2a shows the individual keypoints extracted for the body, face, and hands. Figure 2b illustrates the subset of keypoints used in our model.

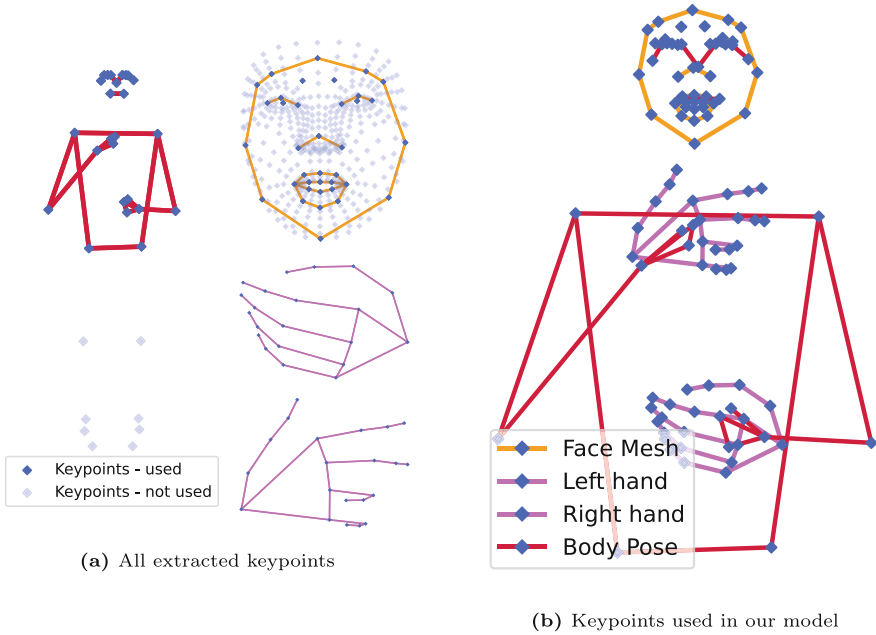


Fig. 2. We use only a subset of the keypoints extracted by MediaPipe. (a) shows all keypoints extracted by the individual MediaPipe models for the body, face, and hands. (b) shows the subset of keypoints that are used as input to our model.

Additionally, we apply normalization to all keypoints. For hand and face keypoints, we use local normalization, which involves creating a square bounding box around them to maintain the aspect ratio and then normalizing them to a range of -1 to 1 . This provides a focused view of facial expressions and hand gestures. For body pose, we use global normalization, where all keypoints are normalized relative to the sign space, ensuring that all keypoints inside the sign space fall within the range of -1 to 1 , see Fig. 3. Global pose normalization provides an overall view of the body pose and the relationships between different body parts.

³ As defined in the YouTubeASL paper [17].

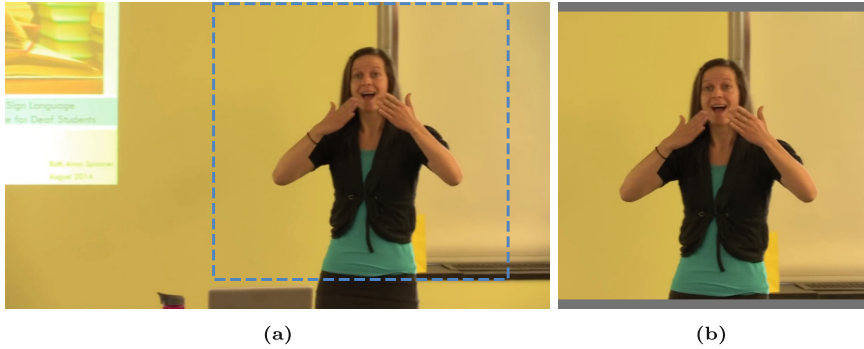


Fig. 3. Video preprocessing based on sign space. (a) illustration of sign space in input frame, (b) cropped and padded frame.

Finally, we crop and pad frames to a square, resize them to a fixed resolution, and save them alongside the extracted keypoints. This preserves aspect ratios while removing unnecessary visual clutter, such as background. Simply resizing large videos to a smaller resolution without cropping the background could result in the loss of fine details in hand and face gestures. To reduce the sequence length, we remove every other frame, resulting in a preprocessed input consisting of 208-dimensional landmark vectors at half the original frame rate.

4.2 Model

Inspired by the YouTubeASL baseline approach, we used a similar, slightly modified version of the T5 [13] encoder-decoder transformer language model. Instead of the traditional approach of using a sequence of textual tokens as the input, we rather embed each 208-dimensional keypoint vector in the encoder using a single learnable linear layer. We experiment with three different T5 architectures: T5-base, T5v1.1-base⁴ for English and mT5-base [19] for both English and Arabic texts.

4.3 Training Pipeline

Our training pipeline follows a two-stage approach: pre-training on the YouTubeASL dataset and fine-tuning on the SSL dataset. This allows the model to first learn general sign language features from a large, diverse dataset (YouTubeASL) and then specialize on the target domain (SSL). The model was evaluated on three distinct test sets.

Pre-training. In the pre-training stage, the model was trained on the YouTubeASL dataset, a large-scale collection of SL videos paired with textual translations. This step enables the model to learn general SL features from a broad

⁴ https://github.com/google-research/text-to-text-transfer-transformer/blob/main/released_checkpoints.md.

set of examples, which are crucial for transferring knowledge to the smaller SSL dataset. Since the original YouTubeASL paper [17] doesn't provide a training and validation split, we randomly sampled our own split with a ratio of 9:1 in such a way that the clips from the same source video can not be in both the training and validation subsets. We initialized the model with pre-trained T5-base weights, originally trained on textual data, and adapted it to process 208-dimensional keypoint vectors by embedding them into the encoder via a linear layer. The sequence-to-sequence framework was employed, where the encoder processed linearly mapped keypoint sequences and the decoder generated textual output. The results of the pre-trained models on the How2Sign dataset can be found in Table 2.

Table 2. How2Sign evaluation of our models pre-trained on YouTubeASL without any further fine-tuning on How2Sign.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4
T5-Base	22.21	3.91	1.49	0.62
T5v1.1-Base	24.52	4.54	1.71	0.72
mT5-Base	25.17	4.78	1.79	0.75

Fine-Tuning. After pre-training, the model was fine-tuned on the SSL dataset to adapt to the specific characteristics of the English American Sign Language from YouTubeASL to the Arabic Saudi Sign Language. Fine-tuning helps bridge the gap between the general features of American and Saudi sign language. During this stage, the model was trained on keypoint-text pairs from the SSL dataset using the same sequence-to-sequence framework.

Evaluation. To evaluate the performance of our model, we examine its robustness and generalization across three distinct test scenarios, as outlined in Table 1. For each testing scenario, we are providing the standard BLEU, BLEURT, and ROUGE-L scores.

5 Experiments

5.1 Pre-training

In the pre-training stage, we have trained T5-base, T5v1.1-base, and mT5-base models for a total of 200,000 training steps. The pre-training stage was conducted using 4 AMD MI250x GPU modules, split into 8 GCDs for each model. The T5-base model demonstrated efficient training with a learning rate of 0.001, while the T5v1.1-base and mT5-base models were trained with a smaller learning rate of 0.0004. For the pre-training, we use an effective batch size of 256 samples. Since mT5-base is a larger model, we use half the per-device batch size and double the gradient accumulation step to fully utilize our GPUs. We use Adafactor to optimize the model's parameters.

5.2 Fine-Tuning

For fine-tuning, we conducted two rounds of experiments using three models trained on the English transcription: T5, T5v1.1, and mT5. The first round used the base model weights, and the second round used the weights pre-trained on YouTubeASL. The mT5 model was fine-tuned twice: once with the original Arabic transcription and once with translated English transcription using Google Translate, similar to T5 and T5v1.1. This resulted in a total of eight experiments. All fine-tuning experiments were conducted on 8 NVIDIA A100-80GB GPUs. The learning rate was set to 0.001 with the AdamW optimizer and a linear LR scheduler, with a batch size of 16 per GPU (128 in total). For mT5, the batch size was reduced to 4, using gradient accumulation of 4 to mitigate memory issues. A weight decay of 0.01 was applied, and the models were trained for 100 epochs.

Table 3. Relevant metrics for different T5 model variants across three test scenarios.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	24.05	9.59	4.96	2.73	−1.23	22.1
T5v1.1-Base	26.16	9.25	4.39	1.59	−1.26	24.33
mT5-Base (Eng)	23.63	8.32	3.98	1.46	−1.21	21.36
mT5-Base (Ar)	10.84	2.99	1.28	0.72	—	11.03

(a) Test-1: Unseen Signers – Unseen Sentences

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	24.46	9.14	4.53	2.01	−1.24	22.22
T5v1.1-Base	26.87	11.25	6.05	2.78	−1.21	25.17
mT5-Base (Eng)	26.72	10.53	5.42	2.79	−1.22	24.44
mT5-Base (Ar)	13.30	4.28	1.64	0.66	—	13.91

(b) Test-2: Seen Signers – Unseen Sentences

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	84.07	81.09	80.59	80.37	0.53	82.67
T5v1.1-Base	88.46	86.27	85.84	85.75	0.66	87.25
mT5-Base (Eng)	87.76	85.62	85.22	85.16	0.64	86.71
mT5-Base (Ar)	85.54	84.27	83.99	83.87	—	85.53

(c) Test-3: Unseen Signers – Seen Sentences

Table 4. Relevant metrics of T5 model variants initialized with YouTubeASL pre-trained weights, evaluated on three test protocols.

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	35.89	17.33	11.14	7.48	-0.89	33.34
T5v1.1-Base	34.76	16.79	10.05	5.56	-0.98	31.5
mT5-Base (Eng)	33.50	16.07	9.77	5.66	-1	30.72
mT5-Base (Ar)	16.75	5.40	1.86	0.81	-	16.79

(a) Test-1: Unseen Signers – Unseen Sentences

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	35.78	17.53	10.34	5.72	-0.89	33.15
T5v1.1-Base	35.59	16.96	9.92	5.23	-0.93	32.68
mT5-Base (Eng)	32.92	14.85	8.52	4.74	-1.02	30.6
mT5-Base (Ar)	18.16	6.37	2.66	1.47	-	17.94

(b) Test-2: Seen Signers – Unseen Sentences

Model	BLEU-1	BLEU-2	BLEU-3	BLEU-4	BLEURT	ROUGE-L
T5-Base	95.17	94.02	93.78	93.67	0.86	94.4
T5v1.1-Base	94.50	93.16	92.83	92.58	0.85	93.97
mT5-Base (Eng)	94.64	93.44	93.19	93.04	0.85	94.09
mT5-Base (Ar)	92.97	92.37	92.42	92.48	-	92.53

(c) Test-3: Unseen Signers – Seen Sentences

5.3 Results

The results of fine-tuning on the original base models’ weights computed using SarceBLEU, BLEURT, and ROUGE-L metrics are shown in Table 3, while the results for fine-tuning on YouTubeASL’s checkpoints are presented in Table 4. Since BLEURT is a trained metric and was not trained on Arabic, the results were not computed for the model fine-tuned on Arabic data. The tables demonstrate a clear increase in the scores across all test sets, supporting our claim that incorporating a large-scale sign language dataset during pre-training enhances the model’s generalization across different and unseen languages. Notably, the results for Test-3 are significantly higher than those of the first two tests, as the sentences in the latter were unseen during training. Additionally, the consistent improvements across BLEU-1 to BLEU-4 and ROUGE-L indicate better word capture and phrase construction, noting the lower BLEU-4 scores compared to BLEU-1, as it focuses on longer phrases.

The results show that mT5 models trained on Arabic transcriptions perform poorly compared to those trained on English. This may be due to mT5 being trained on the mC4 dataset, where the representation of Arabic is smaller than that of English [19]. Additionally, both tables reveal that mT5 models trained on English transcriptions consistently outperform those trained on Arabic. This suggests that translating non-English labels into English during mT5 training could improve model performance.

6 Conclusion

This paper tested the effectiveness of T5-based models for the Saudi Sign Language translation task using a novel dataset. In our experiments, we compared two main training protocols - direct training and pre-training on the large-scale American Sign Language (ASL) dataset. The SSL dataset incorporates three different testing protocols, which allowed us to systematically evaluate generalization across unseen signers and sentences. Additionally, the challenges posed by SSL-specific features, such as face coverings and unique grammatical structures, directly increase translation difficulty. Our results confirm the cross-linguistic transferability of sign language translation models and highlight the effectiveness of leveraging pre-training to overcome data scarcity issues in low-resource sign languages like SSL.

We would like to focus on two main research directions in our future work. Firstly, testing of different input modalities. In this paper, we utilized only pose as an input modality; for example, the DINO or MAE features can also encode relevant information. In fact, we were able to conduct some preliminary experiments with the DINO modality. However, we did not reach any satisfactory results with them. We argue that this can be caused by the fact that the dataset is relatively small, and therefore, models are not able to fully leverage the strength of the deep features.

Secondly, improvements in the preprocessing pipeline. In the current preprocessing pipeline, we entirely omit frames with multiple persons, resulting in less data for the training. Additionally, we would like to test different types of normalization, which seems to play a critical role in the quality of the preprocessing pipeline as demonstrated in [2].

Acknowledgments. The authors with UWB affiliation have been supported by the grant of the University of West Bohemia, project No. SGS-2025-011. Computational resources were provided by the e-INFRA CZ project (ID:90254), supported by the Ministry of Education, Youth and Sports of the Czech Republic. We thank the Saudi Data & AI Authority (SDAIA) for hosting the Winter School, where this work was conducted and for providing generous computing support.

References

1. Altamimi, H.S., Alsager, H.N.: Argument structure and word order in Saudi sign language. *J. Lang. Teach. Res.* **14**(1), 203–214 (2023)

2. Boháček, M., Hruží, M.: Sign pose-based transformer for word-level sign language recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) Workshops, pp. 182–191 (2022)
3. Camgoz, N.C., Hadfield, S., Koller, O., Ney, H., Bowden, R.: Neural sign language translation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
4. Chen, Y., Wei, F., Sun, X., Wu, Z., Lin, S.: A simple multi-modality transfer learning baseline for sign language translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5120–5130 (2022)
5. Duarte, A., et al.: How2sign: a large-scale multimodal dataset for continuous American sign language. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2735–2744 (2021)
6. Gong, J., Foo, L.G., He, Y., Rahmani, H., Liu, J.: Llms are good sign language translators. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 18362–18372 (2024)
7. Gueuwou, S., Siake, S., Leong, C., Müller, M.: Jwsign: A highly multilingual corpus of bible translations for more diversity in sign language processing (2023). <https://arxiv.org/abs/2311.10174>
8. Hu, H., Zhao, W., Zhou, W., Li, H.: Signbert+: hand-model-aware self-supervised pre-training for sign language understanding. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(9), 11221–11239 (2023)
9. Jocher, G., Chaurasia, A., Qiu, J.: Ultralytics YOLO (2023). <https://github.com/ultralytics/ultralytics>
10. Kumar, M., Visagan, S.S., Mahajan, T.S., Natarajan, A.: Enhanced sign language translation between American sign language (ASL) and Indian sign language (isl) using llms (2024). <https://arxiv.org/abs/2411.12685>
11. Liang, H., et al.: Llava-slt: visual language tuning for sign language translation (2024). <https://arxiv.org/abs/2412.16524>
12. Lugaresi, C., et al.: Mediapipe: a framework for perceiving and processing reality. In: Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR), vol. 2019 (2019)
13. Raffel, C., et al.: Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.* **21**(140), 1–67 (2020)
14. Rust, P., Shi, B., Wang, S., Camgöz, N.C., Maillard, J.: Towards privacy-aware sign language translation at scale (2024). <https://arxiv.org/abs/2402.09611>
15. Sprenger, K., Mathur, G.: Observations on word order in Saudi Arabian sign language. *Sign Lang. Stud.* **13**(1), 122–134 (2012)
16. Tanzer, G., Zhang, B.: Youtube-sl-25: a large-scale, open-domain multilingual sign language parallel corpus (2024). <https://arxiv.org/abs/2407.11144>
17. Uthus, D., Tanzer, G., Georg, M.: Youtube-asl: a large-scale, open-domain American sign language-English parallel corpus. *Adv. Neural. Inf. Process. Syst.* **36**, 29029–29047 (2023)
18. Wong, R., Camgoz, N.C., Bowden, R.: Sign2gpt: leveraging large language models for gloss-free sign language translation (2024). <https://arxiv.org/abs/2405.04164>
19. Xue, L.: mt5: a massively multilingual pre-trained text-to-text transformer. arXiv preprint [arXiv:2010.11934](https://arxiv.org/abs/2010.11934) (2020)
20. Yano, C., Fukuchi, A., Fukasawa, S., Tachibana, H., Watanabe, Y.: Multilingual sentence-t5: scalable sentence encoders for multilingual applications. arXiv preprint [arXiv:2403.17528](https://arxiv.org/abs/2403.17528) (2024)

21. Yin, A., Zhong, T., Tang, L., Jin, W., Jin, T., Zhao, Z.: Gloss attention for gloss-free sign language translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 2551–2562 (2023)
22. Zhou, B., et al.: Gloss-free sign language translation: improving from visual-language pretraining. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 20871–20881 (2023)
23. Zhou, H., Zhou, W., Qi, W., Pu, J., Li, H.: Improving sign language translation with monolingual data by sign back-translation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 1316–1325 (2021)

Author Index

A

Achkeev, Andrey A. II-271
Agüero, Marvin M. II-242
Alfaro-Contreras, María II-144
Alharbi, Yazeed II-331
Alhejab, Ali II-331
Ali, Ahmed II-331
Alkalifah, Badriah II-331
Alkanhal, Lamy II-331
Al-Radhi, Mohammed Salah I-52
Alsalihi, Mohammed Hamzah I-295
Ananeva, Anastasia II-203
Ausev, Egor I-307

B

Bakaev, Maxim I-129
Bekiryazıcı, Şule I-333
Bessonitsyn, Evgeny I-322
Blinova, Olga V. II-257
Bobrov, Nikolay I-256, II-318
Bochkarev, Vladimir I-217, II-271
Bogdanova-Beglarian, Natalia V. II-257
Bokkahalli Satish, Shree Harsha I-25
Bondarenko, Ivan II-130
Borzykh, Anna I-267
Boulianne, Gilles II-87
Bouziane, Abdelghani I-157
Bratières, Sébastien II-70
Brovkina, Ekaterina II-231

C

Chien, Pei-Wen II-189
Chirkovskiy, Artem I-307
Christensen, Heidi I-203
Clarke, Jason II-289
Close, George I-39
Colbes, José D. II-242

D

Delić, Vlado II-118
Dobsinszki, Gergely II-60

Dolgushin, Mikhail I-173, II-104
Duret, Jarod II-3

E

Efimova, Valeria I-322
Estève, Yannick II-3
Evdokimova, Vera II-302

F

Fegyő, Tibor II-60
Fivela, Barbara Gili I-241
Fongaro, Marie I-241
Frolova, Olga I-188

G

Gajre, Kunjan I-113
Galametz, Audrey II-3
Ganbaatar, Tsend-Ayush I-256
Gerazov, Branislav II-70
Goetze, Stefan I-39, I-203, II-289
Gonchar, Larisa II-231
Gosztolya, Gábor I-101
Gotoh, Yoshihiko II-289
Grechanyi, Severin I-188
Grillo, Sebastián A. II-242
Gruber, Ivan II-331
Gupta, Rajnidhi I-113
Gupta, Vishwa II-87
Guseva, Daria I-173

H

Hain, Thomas I-39
Halmai, Dániel I-101
Hanılçı, Cemal I-333
Harischandra, Inuri I-83
Haznedaroglu, Ali II-45
Henter, Gustav Eje I-25
Hévr, Gabriel I-241
Hillmann, Stefan II-219
Hong, Kris I-39

Hrúz, Marek II-331
Hsu, Jia-Lien II-189

I

Iurevtseva, Polina I-129
Ivanko, Denis II-174

J

Javorek, Václav II-331

K

Kádár, Máté Soma II-60
Kagirov, Ildar II-104
Karimova, Ekaterina II-318
Karpov, Alexey I-173
Kaya, Heysem I-3
Khokhlova, Maria V. II-257
Khristoforov, Stanislav I-217
Kipyatkova, Irina II-104
Kiseleva, Kseniia II-104
Kleshnev, Egor I-145, I-188
Kobus, Catherine II-3
Kochetkova, Uliana I-67
Koroteeva, Olesia II-231
Kostyuchenko, Evgeny I-228
Kowol, Philine II-219
Kragin, Alexander II-29

L

Laperrière, Gaëlle II-3
Lebedev, Andrei I-188
Legchenko, Anton II-130
Leung, Wing-Zin I-203
López-García, Alejandro II-144
Loukachevitch, Natalia II-29
Lyakso, Elena I-145, I-188

M

Mády, Katalin II-60
Maksimova, Maria II-302
Martin, Marion-Cécile II-3
Maslenikova, Aleksandra S. I-278
Matveev, Anton I-188
Matveev, Yuri II-231
Mdhaftar, Salima II-3

Mello, Julio C. II-242
Meyer, Julien II-144
Mihajlik, Péter II-60
Mitrofanova, Olga I-129
Motovskikh, Leonid I-256, II-318

N

Nayanathara, Sasangi I-83
Németh, Géza I-52
Nersisson, Ruban I-188
Niebuhr, Oliver II-13
Nikolaev, Aleksandr I-188
Nosek, Tijana II-118

O

Oleiwani, Jo II-3
Ozcan, Neyir I-333
Ozkose, Yunus Emre II-45

P

Patil, Hemant A. I-113
Pearsell, Sara M. II-13
Pekar, Darko II-118
Pélissier, Maud I-241
Polevoi, Anton II-29
Politi, Marcello II-70
Popova, Tatiana I. I-278, II-257
Potapov, Vsevolod I-256, II-318
Potapova, Rodmonga I-256, II-318
Pozhilov, Iurii II-318
Purohit, Ravindrakumar M. I-113
Pushpananda, Randil I-83

R

Rahmani, Abdelkader Seif El Islam I-157
Reiss, Joshua II-161
Ryumin, Dmitry II-174

S

Schmück, Samuel II-13
Sečujski, Milan II-118
Shabanov, Petr I-188
Shangina, Ekaterina I-307
Sherban, Anastasiia I-67
Sherstinova, Tatiana Y. II-257

Shevchenko, Tatiana [I-267](#)
Shevlyakova, Anna [I-217](#), [II-271](#)
Shurid, Sadi Mahmud [I-52](#)
Smolskaya, Diana [II-318](#)
Sogancioglu, Gizem [I-3](#)
Stanojev, Vuk [II-118](#)
Straka, Jakub [II-331](#)
Suzić, Siniša [II-118](#)
Székely, Éva [I-25](#)
Sztahó, Dávid [I-295](#)

T

Tirskikh, Danil [II-231](#)
Tomilov, Anton [II-203](#)

V

Valdez, Carlos U. [II-242](#)
Valero-Mas, Jose J. [II-144](#)

Vázquez Noguera, José Luis [II-242](#)
Volkova, Marina [I-307](#), [II-203](#)

W

Weerakoon, Thamira [I-83](#)
Whetten, Ryan [II-3](#)

X

Xu, Zhiyuan [II-161](#)

Y

Yahiaoui, Yasser [I-157](#)
Yakovenko, Anton [I-322](#)

Z

Zaburdaev, Alexander [II-174](#)
Zaslavskiy, Mark [I-322](#)
Železný, Tomáš [II-331](#)