Word Sense Induction in Russian: Evaluation of Corpora Pre-processing Techniques and Model Selection

Kondratenko Yana and Mitrofanova Olga

Saint-Petersburg State University, Saint-Petersburg, Russia
kondratenkoyak@yandex.ru, o.mitrofanova@spbu.ru

Abstract. The article proposes the solution to the problem of word sense induction (WSI) in Russian. WSI refers to the task of resolving ambiguity by deriving features from corpora without using knowledge bases, dictionaries and predefined lists of word senses. This study examines the impact of data preprocessing on the performance of WSI techniques using clustering algorithms for context vector representations based on a series of BERT models for Russian language. Various methods of data preprocessing were analyzed, including tokenization, lemmatization, stop words removal and selection of pre-trained models of context vector representations. Experiments have shown that the presence or absence of lemmatization does not affect clustering results, while other factors such as the removal of stop words and the choice of data vectorization model can significantly affect clustering. The results of the study can be applied in the procedures of semantic annotation of text corpora.

Keywords: Word Sense Induction, Russian text corpora, Distributional semantic models, BERT, data preprocessing.

1 Introduction

Ambiguity is an immanent property of a natural language text: texts generated with the help of a finite set of lexical units and morpho-syntactic rules for combining them are potentially intended to describe the infinite content of extralinguistic reality. Ambiguity, in this case, can be caused both by the multifunctionality of linguistic mechanisms and items and by the similarity or contiguity of information transmitted by language. In terms of communication, the speaker and the listener choose the combination of conceptual features of texts and the linguistic means, which is most likely related to a particular speech situation and in a specific speech context [1-3]. At the same time, resolvable and unresolvable ambiguity cases should be contrasted. Restrictions on the scope of disambiguation can be associated with communicative functions of speech acts (e.g., puns, allegorical statements, etc.) and with the essential impossibility of choosing a single interpretation from a set of acceptable ones in a given context (e.g., diffusion of the meaning of a word or expression, limited context, etc.) [4-6]. The causes of ambiguity are diverse in nature; it is customary to talk about morphological, syntactic, lexical-semantic, pragmatic ambiguity. As a rule, ambiguity manifests itself simultaneously at several levels of context representation, and this complicates modeling of this phenomenon in NLP.

The reproduction of natural mechanisms of ambiguity resolution in language processing is of research interest in most projects dealing with automatic text understanding systems. This reproduction is based on generalized structural models which reflect the observed linguistic entities and relationships in an abstract form far from linguistic reality. The given divergence is the focus of formal procedures for text analysis, synthesis and transformation. Therefore, automatic text understanding systems offer a solution to the problem of ambiguity in the form of a set of conventional procedures applied at separate levels (morphological, syntactic, semantic, etc.).

The following procedures can be applied in the studies of lexical-semantic ambiguity: 1) automatic lexical-semantic annotation, e.g. [7-9]; 2) lexical-semantic differentiation of a meaning of a word or expression in contexts, e.g. [10-11]; 3) automatic inference of the meaning of a word or expression from contexts, e.g. [12-13]. All these approaches are based on the assumption that ambiguities may be registered in the text. In our work we follow the assumption on the joint analysis of polysemy and homonymy which are weakly differentiated in computational semantic procedures (cf. the ideas of semantic annotation and disambiguation in the Russian national corpus (RNC) [7]).

Lexical-semantic annotation solves the problem of correlating a linguistic expression with a certain category or group of categories, however, the markup itself (for example, a group of semantic tags) does not allow to represent the meaning of a word or expression completely. The markup performs the function of identifying and differentiating meanings rather than their detailed description. A text with lexical-semantic markup does not contain all possible tags for describing a particular meaning, but only those that, in combination with a lemma or contextual markers, make it possible to differentiate meanings of a polysemous word. The consistency of lexical-semantic markup as a procedure is based not only on the fact that it is possible to compose an adequate markup scheme, but also on the fact that the text itself can be reduced to its unambiguous representation. Semantic annotation does not necessarily imply total ambiguity resolution (e.g., if an unambiguous fragment of the corpus is marked), but it also does not exclude it (e.g., in the case of assigning all possible combinations of tags to a word or expression that is ambiguous in a particular context).

The procedure for automatic word sense disambiguation (WSD) is based on dictionary data and/or context markers of different types and is applied precisely to those parts of the corpus that, when annotated, can receive an ambiguous interpretation. The initial hypothesis on which this procedure is formed is related to the fact that filiations of lexical meanings are pre-defined in the dictionary as well as the elements of contexts associated with these meanings are identified. WSD is carried out in the course of classifying contexts with respect to lexical-semantic patterns which store features that are characteristic of the use of a word in a particular meaning.

The procedure for deriving word meanings from the WSI corpus is based on the assumption of the ambiguity of words in the lexicon and at the same time it is possible due to another assumption — that semantic ambiguity in the corpus is removed by linguistic insights (context) or algorithmically. During WSI, contexts or context elements are clustered, and the resulting clusters are identified with values. For example, if there are the following contexts of the polysemous word *лук* (*onion/bow*): as a result of the task execution, the contexts should be grouped as follows: contexts 1, 4 and 5 will fall into one cluster, and 2 and 3 — into another.

1. *Лицо женщины было испуганным, из груди торчала красная стрела, а над головой женщины, в небе, летал голый маленький мальчик с*

крылышками, с ***луком*** *в руках, и специальными черточками было нарисовано, что тетива на* ***луке*** *дрожит. [Евгений Гришковец. ОдноврЕмЕнно (2004)] (The woman's face was scared, a red arrow was sticking out of her chest, and a naked little boy with wings was flying in the sky above the woman's head, with a* ***bow*** *in his hands, and it was drawn with special dashes that the bowstring was trembling..)*

2. *А он рисовал эту картину, у него мерзли руки, и он грел свои руки…, дышал на них…, а изо рта, возможно, пахло* ***луком****, потому что он поел* ***луку****… [Евгений Гришковец. ОдноврЕмЕнно (2004)] (And he painted this picture, his hands were freezing, and he warmed his hands ..., breathed on them ..., and his breath probably smelled of* ***onions****, because he ate* ***onions…****)*
3. *Частым блюдом была фасоль с* ***луком*** *и постным маслом. [Эдуард Лимонов. У нас была Великая Эпоха (1987)] (A frequent dish was beans with* ***onions*** *and vegetable oil.)*
4. *Стреляли птиц из самодельных* ***луков****. Кое-кто ловил в море рыбу наволочкой. [И. Грекова. Фазан (1984)] (They shot birds from homemade* ***bows****. Someone was fishing in the sea with a pillowcase.)*
5. *Я не удивился, если бы вдруг тут сию минуту увидел запыленный пурпуровый плащ выходящего из каменной щели кудрявого бога в венке из виноградных листьев, с убитой серной на плече, с колчаном и* ***луком*** *за спиной, с кубком молодого вина в руке — прекрасного и слегка во хмелю, как сама поэзия, которая его породила. [В. П. Катаев. Алмазный мой венец (1975-1977)] (I wouldn't be surprised if I suddenly saw a dusty purple cloak of a curly-haired god emerging from a stone crack in a wreath of grape leaves, with a dead chamois on his shoulder, with a quiver and a* ***bow*** *behind his back, with a goblet of young wine in his hand - beautiful and slightly hops, like the very poetry that gave birth to him.)*

Lexical-semantic disambiguation is one of the tasks of automatic language processing which is to be solved to improve results in the fields of machine translation, question-answering systems, and information extraction [14]. Current baselines providing effective decisions of several tasks (supervised – knowledge-based, monolingual – multilingual, coarse-grained – fine-grained, all-words – target words disambiguation, WSD – WSI, etc.) were worked out for English and some other languages in course of SemEval competition series [15]. Russian data was thoroughly investigated within RUSSE competition [16] and semantically annotated corpora development [17]. The Russian language has got rich morphology, and, in this regard, text preprocessing can have an impact on the result of context clustering.

The purpose of this work is to consider the influence of data analysis (tokenization, lemmatization, punctuation marks removal, vector representation) on the results of automatic sense induction in Russian. We focus our attention on the choice of contextualized distributional embedding models, preprosessing techniques and clustering algorithms which provides an increase in the quality of ambiguity resolution and makes up for the lack of knowledge in this field of research.

2 Related work

WSD is the subject of many studies, the first of which appeared in the 1960s. The majority of modern solutions are based on machine learning and statistical methods. Marking up data, compiling annotated corpora and creating inventories of values for disambiguation is an extremely resource-intensive task, and therefore methods of unsupervised WSD have become very popular.

The core ideas of this method are exposed in several statements: a) each ambiguous word is represented as a set of context vectors with a given word, b) the contexts for each ambiguous word are grouped into clusters using one of the clustering methods, c) for each cluster there is a centroid, on the basis of which further disambiguation is performed for new examples of the use of the target word.

Within the competition SemEval 2013 solutions to the problem for the English language were presented. The participants were asked to cluster the search results for an ambiguous query into semantically related groups in accordance with their values. The data set consisted of one hundred ambiguous queries with a length of no more than four words and 6400 results [18]. Several models were presented in the competition, including [19-21].

To solve this problem for the Russian language in 2018, the RUSSE’2018 competition was held as part of the Dialogue conference [22]. Participants presented clustering models trained on three data sets containing ambiguous words and contexts of their use. The models presented at the competition are described in the works [23-24].

The results of the competition showed that modern systems still cope with the task of deriving values from contexts with great difficulty on the material of polysemants with high detail of values, however, at the same time, they show high results for homonyms (wiki-wiki corpus).

For example, [23] presented a solution based on clustering of «semantic fingerprints» of contexts using the Affinity Propagation algorithm. So, as a result, the highest ARI score reached 0.77 for the wiki-wiki corpus.

Later, this algorithm was modified by using the BERT model to obtain vector representations [26], which allowed increasing the accuracy to an ARI value equal to 0.81.

In another paper [24], a similar algorithm is considered, vector representations of the context in which were calculated as an average weighted vector from vector representations of words obtained using word2vec. In addition, other clustering algorithms were used: in addition to Affinity Propagation, experiments were carried out using DBSCAN, OPTICS, Spectral clustering and Agglomerative clustering algorithms. The maximum accuracy on the wiki-wiki corpus was 0.81 ARI value.

Many papers devoted to the problem of disambiguation describe data preprocessing, but the authors rarely justify the choice of their method. For example, in the works [19-21, 23, 25] words in contexts are preliminarily lemmatized, while in the work [26] there is no lemmatization.

Contextualized word embedding models based on BERT show high efficiency in obtaining semantic vectors of linguistic units on the material of the Russian language, which is confirmed in the works [19, 26]. Such models are based on the BPE (Byte-

Pair Encoding) algorithm, so it is considered that lemmatization is not an obligatory step in data preprocessing when using these models.

The study [27] considers the influence of lemmatization on the results of the problem of disambiguation using ELMo embeddings. The experiments have shown that the absence of lemmatization did not affect the classification accuracy for English, while there was a small but stable increase in accuracy for Russian. The authors suggest that this is due to the rich morphology of Russian. Our experiments must confirm or refute the observations on WSI conditions for Russian.

3 Experiments on WSI

3.1 Research corpus

The experiment described in this study was performed on the Russian data set prepared for the Dialogue Evaluation RUSSE’2018 competition devoted to WSI [28]. The choice of the given dataset is justified by the fact that it was developed as a gold standard for WSD/WSI procedures in Russian. The dataset was involved in evaluation of static distributional semantic models, but the full-scale research for a set of contextualized embedding models haven’t been not performed yet. Thus, our work fills in the gap. The corpus for training and testing WSI algorithms was compiled on the basis of Wikipedia data. It contains 9 polysemous/homonymous nouns (*'бор' (pine forest/boron)*, *'суда' (ships/court)*, *'лук' (onion/bow)*, *'замок'(castle/lock)*, *‘банка’ (bank/pot)*, *‘бит’ (bit/beat)*, *‘горе’ (grief/mountain)*, *‘граф’ (Earl/graph)*, *‘душ’(shower/soul)*) as target words, as well as contexts containing target words used in one of the two meanings provided. In total, 1056 contexts are included in the corpus (some contexts may include several occurrences of the target word). The contexts in the corpus are preprocessed — all digits are removed and all characters are reduced to lowercase. Data example is given in Table 1.

Table 1. Data example.

index	context_id	word	gold_sense_id	predict_sense_id	position	context
1	2	замок	1	NaN	11-16, 17-22, 188-193	*шильонский* ***замок замок*** *шильйон ( ) , известный в русскоязычной литературе как шильóнский* ***зáмок*** *, расположен на швейцарской ривьере , у кромки женевского озера , в км от города монтре .* ***замок*** *представляет собой комплекс из элементов разного времени постройки . (the chillon castle ( ) , known in russian literature as the castle of chillon , is located on the swiss riviera , at the edge of lake geneva , km from the*

						city of montreux . the castle is a complex of elements of different construction times .)
2	3	замок	1	NaN	299-304	*проведения архитектурно - археологических работэстонским реставрационным управлением под руководством архитектора х . и . потти , искусствоведа е . а . кальюнди и при научной консультации доктора исторических наук п . а . рапопорта . с года музей называется государственным музеем выборгский **замок** .(carrying out architectural and archaeological works by the estonian restoration department under the supervision of architect h. i. potti , art critic e . a. kaliundi and with the scientific advice of the doctor of historical Sciences p. a. rapoport . since the year the museum has been called the vyborg **castle** state museum .)*
3	4	замок	1	NaN	111-116	*топи с . , л . белокуров легенда о завещании мавра с . , н . юсупов день рождения с . , р . янушкевич янтарный **замок** с . . (topi s. , l. belokurov legend of the will of the moor s. , n. yusupov birthday s. , r. yanushkevich amber **castle** S. .)*

Unlike the rest of the corpora presented in this competition, the wiki-wiki corpus contains mostly homonymous words, not polysemous ones. In this case the meanings of words are more clearly distinguished, so automatic sense induction is more justified in this experimental setting.

3.2 Experimental setup

Our experiment aims to determine the impact of the following data processing factors on WSI results:

- removal of punctuation marks;
- tokenization (Python libraries NLTK [29], Stanza [30], Razdel [31], Segtok [32], Spacy [33], Moses [34]);

- lemmatization (cf. Python libraries above);
- removing duplicate words;
- context embeddings (transformers ruBERT-tiny [35], LaBSE [36-37], RuBERT [38]).

Distributed vector representations for WSI were obtained on the basis of pre-trained BERT models for sentence embeddings. The input contexts pre-processed in various ways were clustered using the following clustering algorithms: KMeans, Affinity Propagation, DBSCAN, OPTICS. The range of algorithms was expanded in comparison with RUSSE protocols, thus, we obtain novel results concerning the choice of clustering techniques suitable for WSD/WSI. The results were evaluated by using an Adjusted Rand Index.

Tokenization. To consider the impact of tokenization, several existing algorithms implemented within libraries for natural language processing were selected. The following libraries were selected: NLTK, Stanza, Razdel, Segtok, Spacy, Moses [29-34]. The difference in the accuracy of these tokenizers was analyzed within the framework of the Naeval project [39] while developing a tool for natural language processing Natasha [40]. As part of the study, differences in the tokenization of the experimental data set using these libraries were analyzed.

- **Features of the processing of the stress sign (´).** The NLTK and Spacy tokenizers highlight an accent if it is on the last letter of a word. The Segtok and Moses tokenizers always separate an accent mark into a separate token. If this symbol is in the middle of a word, it is divided into three tokens according to stress (e.g., *междунаро, ´, дного*). The Razdel tokenizer never allocates an accent mark into a separate token.
- **Features of processing words written with a slash (/).** All tokenizers, except for NLTK and Stanza, allocate the slash sign into a separate token, Stanza divides such words into two tokens, the slash is part of the second token (for example, *км (km), /ч (/h)*), NLTK defines them as one token.
- **Features of processing the degree sign (°).** The NLTK and Razdel tokenizers do not separate this character into a separate token, the other tokenizers do.
- **Features of processing time intervals written with a dash (for example, xi—xii).** The Segtok, Spacy and Moses tokenizers allocate the dash sign as a separate token, in other cases such an entry is processed as a single token.
- **Features of processing some characters.** The Spacy tokenizer allocates a non-breaking space as a separate token, replacing it with the html code of the given character. Moses replaces the meaning of some characters with their html code during tokenization (e.g., quotation marks, square brackets, apostrophe sign).

Lemmatization. As part of the experiment, four processing options were considered: corpus data with Pymorphy2 [41] lemmatization without stop words removal, with Pymorphy2 lemmatization with stop words removal, without lemmatization without stop words removal, without lemmatization with stop words removal. A list of words based on Yandex Wordstat [42] was used as a stop-word dictionary.

Removing duplicate words. During preprocessing, two options for representing contexts were considered. In the first case, contexts were represented as a list of unique words included in the sentence, in the second case, all occurrences of words were saved.

Embeddings. The embedding method used in this study is aimed at obtaining a contextualized embedding of the whole sentence. According to our assumption, different meanings of target words are implemented in contexts whose vectors will be further apart than the vectors of contexts in which a polysemous word is used in the same meaning.

BERT is a language model which is defined as a neural network encoder based on the transformer architecture. When calculating the embedding of a language unit, the model takes into account the right and left context.

In this experiment, we use contextualized models of vector representations. When using such models, the context can be represented in two ways:

- sentence embedding ;
- target word embedding.

To obtain a vector representation of contexts, the following pre-trained BERT models were used:

- **ruBERT-tiny** [35]: this model represents a sentence in the form of a vector with a dimension of 312; BERT-multilingual model was taken as the basis, additional training was carried out on the texts of parallel corpora from Yandex.Translate [43], OPUS-100 [44] and Tatoeba [45];
- **LaBSE** [36-37]: Language-agnostic BERT Sentence Embedding model supports 109 languages, representing the sentence as a vector of 768 dimensions;
- **RuBERT** [38]: Russian BERT model trained on Russian-language Wikipedia and news data represents the context as a vector of 768.

Clustering algorithms. This study was conducted using several clustering algorithms: KMeans, Affinity Propagation, DBSCAN, OPTICS [46].

- **KMeans** is a stochastic algorithm which requires a predetermined number of clusters, which can be a limitation for its use and opposes it to the other three algorithms, which allow not to set the number of clusters in advance but calculate it dynamically. In addition, KMeans algorithm is sensitive to the choice of initial cluster centroids which are initiated at random.
- **Affinity Propagation** is based on the idea of evaluating message passages between data points and requires damping (from 0.5 to 1, by default 0.5) and preference (by default None) as hyperparameters. In our experiment we considered the values 0.5, 0.6, 0.7, 0.8 and 0.9 for the damping parameter and values from -6 to 20, including None for the preference parameter. The final evaluation of the algorithm quality was considered as the average of the algorithm results with values of 0.6 and 0.7 for the damping parameter and None for the preference parameter, since the algorithm with these parameters showed the best results.
- **DBSCAN** algorithm performs density-based spatial clustering of noisy data. It requires ϵ and min_samples value selection. In our case, values from 0.1 to 1 for the ϵ parameter and values from 2 to 9 for min_samples were considered.

These parameters regulate cluster density. The final results were considered as the average between the clustering score with values of 2 and 5 for the min_samples parameter and a value of 0.1 for the eps parameter.

- **OPTICS** algorithm is similar to DBSCAN, but it allows detecting meaningful clusters in data of varying density. In experiments with OPTICS clustering, values for the min_samples parameter from 2 to 15 were considered, the final results were considered as the average of the performance estimates of algorithms with values of this parameter equal to 5 and 8.

Evaluation metrics. The Adjusted Rand Index was applied to evaluate clustering results. The Rand Index (*RI*) of clustering *C* is a measure of clustering agreement that determines the percentage of correctly distributed pairs of elements in two clusterings *C* and *G*. *RI* is calculated by the formula:

$$R(C,G) = \frac{TP + TN}{TP + FP + FN + TN},$$

where *TP* is the number of true positives, i.e., pairs of elements that are in the same cluster both in clustering *C* and in clustering *G, TN* is the number of true negatives, i.e., pairs that are in different clusters in both clusters, and *FP* and *FN*, respectively, the number of false positives and false negatives. *RI* ranges from 0 to 1, where 1 indicates a full match of clusters up to a permutation.

The Adjusted Rand Index (*ARI*) is a modification of the Rand Index that adjusts the *RI* for a random match and makes it vary as expected:

$$ARI(C,G) = \frac{RI(C,G) - E\big(RI(C,G)\big)}{maxRI(C,G)},$$

where *E*(*RI*(*C*,*G*)) is the expected value of *RI*.

Thus, the Adjusted Rand Index has a value close to 0.0 for random labeling regardless of the number of clusters and samples, and exactly 1.0 when the clusters are identical (before permutation). The quality scores of each algorithm were calculated as the average of *ARI* over all target words in the dataset.

4 Experimental results

In course of experiments we managed to reveal the influence of various data preprocessing factors on the results of clustering context embeddings, cf. Table 2. Experiments have shown that lemmatization does not have a stable positive effect on clustering results; on the contrary, quite often it affects negatively. So, in 42% of cases, lemmatization has a positive effect, in 57% − negative, and in 1% it does not affect the results of clustering in any way. The biggest increase is 0.30 points (*ARI* from 0.37 to 0.68). The average ARI value for all experiments without lemmatization is 0.40, with UD-Pipe lemmatization is 0.40, with Pymorphy2 lemmatization is 0.39. The highest results are obtained by experiments with the KMeans algorithm: the average ARI value with Pymorphy2 lemmatization is 0.70, with UD-Pipe lemmatization is 0.70, without lemmatization is 0.71

At the same time, when using word embedding, the effect of lemmatization is much more distinct — in most cases, lemmatization affects the results negatively. Thus, the highest increase in accuracy is 0.47 points (0.40 when using lemmatization Pymorphy2, 0.87 — without lemmatization). The highest results are obtained by experiments with the KMeans clustering algorithm, for example, the average ARI value with Pymorphy2 lemmatization is 0.69, with UD-Pipe lemmatization is 0.78, without lemmatization is 0.88.

Table 2. Estimating the accuracy of algorithms with and without lemmatization.

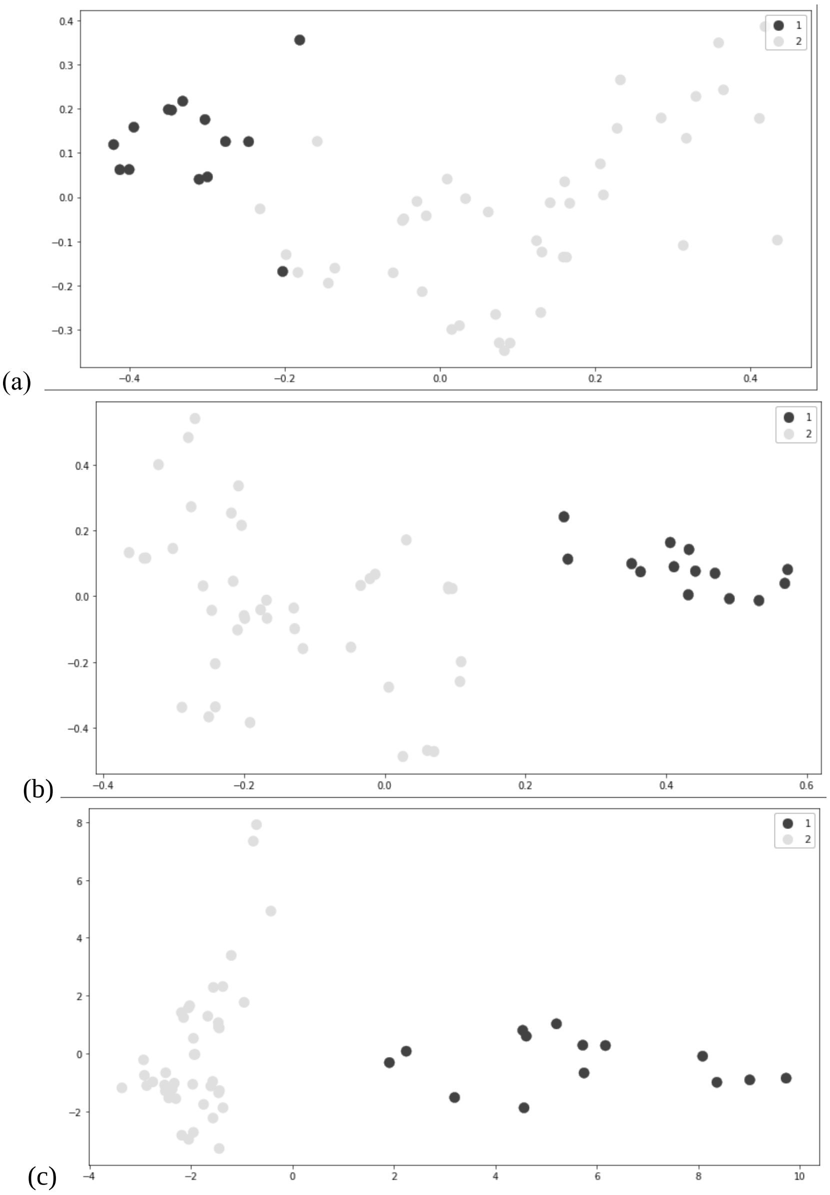
Clustering algorithm	**With lemmatization**				**Without lemmatization**			
	RuBERT-tiny	**LaBSE**	**RuBERT sentence**	**RuBERT word**	**RuBERT-tiny**	**LaBSE**	**RuBERT sentence**	**RuBERT word**
	NLTK tokenizer, without removing punctuation, only unique words, without removing stop words							
KMeans	0.333	**0.785**	0.325	0.438	0.348	0.778	0.332	**0.848**
Affinity Propagation	0.120	**0.214**	0.167	0.060	0.125	0.221	**0.245**	0.073
DBSCAN	0.148	0.447	**0.018**	0.0	0.185	**0.511**	-0.012	0.0
OPTICS	0.073	**0.174**	0.081	-0.032	0.108	0.160	**0.156**	0.148
	NLTK tokenizer, with the removal of punctuation, not only unique words, without removing stop words							
KMeans	**0.401**	0.524	**0.613**	**0.713**	0.326	0.790	0.381	**0.977**
Affinity Propagation	0.122	**0.230**	0.198	0.070	0.142	0.227	**0.209**	0.070
DBSCAN	0.163	**0.475**	-0.015	0.0	0.156	**0.381**	-0.005	**0.0**
OPTICS	0.072	**0.206**	0.120	-0.034	0.069	0.143	**0.222**	0.168

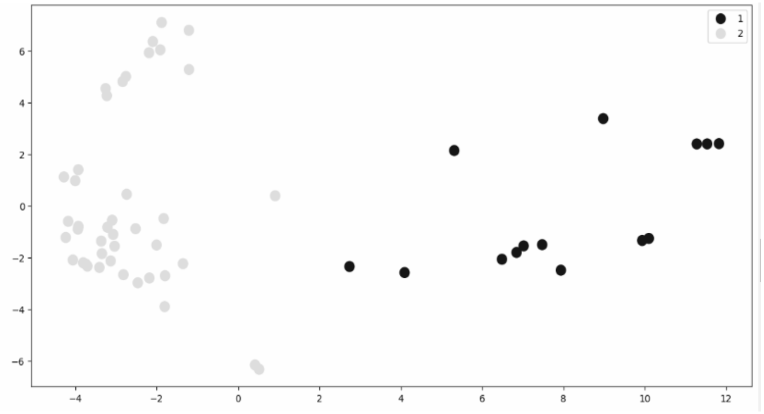
Removing stop words in most cases (62%) shows an increase in clustering accuracy, in 3% of cases it has no effect. The most stable option is preprocessing with the removal of stop words and without lemmatization — such preprocessing does not guarantee the highest performance, but it shows a low result less often than others. The configuration of the Affinity Propagation algorithm and the LaBSE model consistently shows the highest result in lemmatization and removal of stop words compared to other preprocessing options.

Removing punctuation in 55% of cases has a positive effect on clustering results, in 43% it has a negative effect, and in 2% of cases it has no effect. The most significant increase is 0.29 points. The removal of punctuation has a particularly significant effect on the configuration of the DBSCAN algorithm and the RuBERT embeddings — the indicators are consistently higher in experiments in which punctuation is not removed.

Among the methods of vector representation of contexts, the LaBSE model showed the highest results.

Figure 1 shows graphs of context vectors for the word *бор (pine forest / drill)* obtained using different vector representation models.


(a)
0.4
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
−0.4
−0.2
0.0
0.2
0.4
● 1
● 2
(b)
0.4
0.2
0.0
−0.2
−0.4
−0.4
−0.2
0.0
0.2
0.4
0.6
● 1
● 2
(c)
8
6
4
2
0
−2
−4
−2
0
2
4
6
8
10
● 1
● 2



(d)

Fig. 1. Graphs of context embeddings for the word *бор*, embeddings obtained using (a) ruBERT-tiny, (b) LaBSE, (c) RuBERT sentence embeddings, (d) RuBERT word embeddings, all other parameters being equal. The gold standard labels are marked in color.

Clustering algorithms for any preprocessing options in 87% of cases showed the best results on data vectorized using the LaBSE model, in the remaining 13% using RuBERT sentence embedding. Thus, the average ARI value when using the LaBSE model is 0.52, when using the RuBERT sentence embedding it’s 0.46, when using RuBERT word embedding it’s 0.33, and the ruBERT-tiny model — 0.27. Table 3 presents the results of the algorithms for pre-tokenization using NLTK, with the removal of punctuation and stop words, and with lemmatization.

Table 3. Estimating the accuracy of algorithms. Pre-tokenization: NLTK, removal of punctuation and stop words, lemmatization with Pymorphy2.

Clustering algorithm	**ruBERT-tiny**	**LaBSE**	**RuBERT**	
KMeans	0.3537	**0.7956**	0.6025	0.6292
Affinity Propagation	0.1280	**0.2526**	0.1952	0.0665
DBSCAN	0.1635	**0.4384**	-0.0210	0.002
OPTICS	0.07978	**0.2513**	0.2111	-0.0365

The highest results are shown by the configuration of the LaBSE model and the KMeans clustering algorithm (*ARI* from 0.5245 to 0.8074). The next most effective are the DBSCAN clustering algorithm and the LaBSE vector representation (*ARI* from 0.3751 to 0.5354) and the KMeans clustering algorithm and the RuBERT model (*ARI* from 0.3179 to 0.6203).

The highest result was shown by the system implemented using NLTK/Stanza/Razdel/Segtok/Moses tokenizers, without lemmatization, without removing stop words, removing punctuation marks and duplicate words with the KMeans clustering algorithm — the average ARI value for all words of the corpus was 0.97 (up to 1.0 on individual words). However, word embeddings only perform well with the KMeans clustering algorithm, which requires a number of clusters as input. The configuration with the KMeans algorithm also shows the highest result for sentence embeddings: the average corpus ARI value is 0.82 (up to 1.0 on individual words) with any tokenizers, with Pymorphy2 lemmatization, with stopword removal, without punctuation removal, with or without removing duplicates.

5 Conclusion

In this paper we discussed the influence of data analysis on the results of automatic sense induction for Russian. Experiments were carried out to reveal the impact of tokenization, lemmatization, punctuation marks, duplicates, etc. on WSI. Embeddings for WSI were obtained on the basis of pre-trained BERT models: ruBERT-tiny, LaBSE, RuBERT. The input contexts were clustered by KMeans, Affinity Propagation, DBSCAN, OPTICS algorithms. The results were evaluated by an Adjusted Rand Index. The implemented systems showed results exceeding the accuracy of existing systems tested on the same data described in [23, 24, 26].

Table 4. Estimating the accuracy of algorithms

	ARI score for wiki-wiki corpus
Kutuzov, A.[23]	0.77
Arefyev, N., Ermolaev, P., Panchenko, A [24]	0.81
Slapoguzov, A., Malyuga, K., Tsopa, E. [26]	0.81
RUSSE'18 baseline [22]	0.62
Our system	**0.97**

Experiments have shown that lemmatization overall does not improve WSI results, while removing stop words and punctuation provides an increase of ARI. We found optimal configurations for WSI as regards the choice of embedding models and clustering techniques: LaBSE model and KMeans clustering showed the highest results.

Our next work deals with expansion of experiments to all-words WSD and working out a flexible procedure of semantic annotation.

Acknowledgements. Research is performed with partial support of SPbSU Scientific Project № 75254082 «Modeling the communicative behavior of residents of a Russian metropolis in the socio-speech and pragmatic aspects with the use of artificial intelligence methods», RSF grant № 21-78-10148 «Modeling the meaning of a word in individual linguistic consciousness based on distributive semantics».

References

1. Apresyan, Yu.D.: Selected works. Volume 1. Lexical Semantics. Moscow (1995)
2. Zaliznyak, Anna A.: The phenomenon of polysemy and the ways of its description. In: Issues of linguistics. M., 2004. № 2. P. 20-45. (1995)
3. Vinograd, T.: Understanding Natural Language. Academic Press (1972)
4. Apresyan, Yu.D.: On Regular Polysemy. In: Proceedings of the Academy of Sciences of the USSR. Department of Literature and Language. Vol. XXX. Issue 6. M. P. 509-523. (1971)
5. Kopotev, M.: Introduction to Corpus Linguistics. Prague. (2014)
6. Shmelev, D.N.: Essays on the Semasiology of the Russian Language. Moscow. (1964)
7. Rakhilina, E.V., Kustova, G.I., Lyashevskaya, O.N., Reznikova, T.I., Shemanaeva, O.Yu.: The Tasks and Principles of Lexical-Semantic Annotation in RNC // The Russian National Corpus: 2006-2008. New Results and Perspectives. St.-Petersburg: Nestor-Istoriya. P. 215-239. (2009)
8. Mudraya, O.V., Babych, B.V., Piao, S., Rayson, P., Wilson, A.: Developing a Russian Semantic Tagger for Automatic Semantic Annotation. In: Proceedings of the International Conference «Corpus Linguistics-2006». St.-Petersburg, Russia. P. 290-297. (2006).
9. Palmer, M., Kingsbury, P., Gildea, D.: The Proposition Bank: An Annotated Corpus of Semantic Roles. In: Computational Linguistics. 31 (1): 71–106. (2005)
10. Word Sense Disambiguation: Algorithms and Applications. Eds.: E. Agirre, P. Edmonds. Springer, New York, NY. 2006.
11. Navigli, R.: Word Sense Disambiguation: A Survey. ACM Computing Surveys. 41(2). P. 1–69. (2009)
12. Nasiruddin, M.: A State of the Art of Word Sense Induction: A Way Towards Word Sense Disambiguation for Under-Resourced Languages (PDF). TALN-RÉCITAL 2013. Les Sables d'Olonne, France. P. 192-205. (2013)
13. Panchenko, A., Lopukhina, A., Ustalov, D., Lopukhin, K., Arefyev, N., Leontyev, A., Loukachevitch, N.: RUSSE'2018: a Shared Task on Word Sense Induction for the Russian Language. In: Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference «Dialogue». P. 547-564. RSUH, Moscow. (2018)
14. Moro, A., Navigli, R.: SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In: Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval-2015). P. 288-297. Denver, Colorado. Association for Computational Linguistics (2015)
15. URL: https://en.wikipedia.org/wiki/SemEval
16. URL: https://russe.nlpub.org/2018/wsi/
17. Bolshina, A.S., Loukachevitch, N.V.: Weakly supervised word sense disambiguation using automatically labelled collections. In: Trudy ISP RAN/Proc. ISP RAS. Vol. 33. Issue 6. P. 193-204. (2021)
18. Navigli, R., Vannella, D.: SemEval-2013 Task 11: Word Sense Induction and Disambiguation within an End-User Application. In Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), pp. 193–201, Atlanta, Georgia, USA. Association for Computational Linguistics (2013)
19. Amrami, A., Goldberg, Y.: Towards Better Substitution-Based Word Sense Induction. In: ArXiv. URL: https://arxiv.org/abs/1905.12598 (2019)
20. Amplayo, R.K., Hwang, S., Song, M.: Autosense Model for Word Sense Induction. In: Proceedings of the AAAI Conference on Artificial Intelligence. Vol. 33. № 1. P. 6212-6219. (2019)

21. Amrami, A., Goldberg, Y.: Word Sense Induction with Neural biLM and Symmetric Patterns. In: ArXiv. URL: https://arxiv.org/pdf/1808.08518.pdf (2018)
22. Panchenko, A., Lopukhina, A., Ustalov, D., Lopukhin, K., Arefyev, N., Leontyev, A., Loukachevitch, N.: RUSSE'2018: a shared task on word sense induction for the Russian Language. In: ArXiv. URL: https://arxiv.org/vc/arxiv/papers/1803/1803.05795v2.pdf (2018)
23. Kutuzov, A.: Russian Word Sense Induction by Clustering Averaged Word Embeddings In: ArXiv. URL: https://arxiv.org/ftp/arxiv/papers/1805/1805.02258.pdf (2018)
24. Arefyev, N., Ermolaev, P., Panchenko, A.: How much does a word weigh? Weighting word embeddings for word sense induction. In: Computational Linguistics and Intellectual Technologies: Proceedings of the International Conference «Dialogue-2018». Moscow. P. 68-84. (2018)
25. Eyal, M., Sadde, S., Taub-Tabib, H., Goldberg, Y.: Large scale substitution-based word sense induction. In: ArXiv. URL: https://arxiv.org/abs/2110.07681 (2022)
26. Slapoguzov, A., Malyuga, K., Tsopa, E.: Word Sense Induction for Russian Texts Using BERT. In: Conference of Open Innovations Association. FRUCT. №. 28. P. 621-627. (2021).
27. Kutuzov, A., Kuzmenko, E.: To Lemmatize or Not to Lemmatize: How Word Normalisation Affects ELMo Performance in Word Sense Disambiguation. In: Proceedings of the First NLPL Workshop on Deep Learning for Natural Language Processing. Turku, Finland. Linköping University Electronic Press. P. 22–28. (2019).
28. URL: https://russe.nlpub.org/2018/wsi/
29. URL: https://www.nltk.org/
30. URL: https://stanfordnlp.github.io/stanza/Stanza
31. URL: https://github.com/natasha/razdel?ysclid=lgmjj9t4cn752154674
32. URL: https://pypi.org/project/segtok/
33. URL: https://spacy.io/
34. URL: https://pypi.org/project/mosestokenizer/
35. URL: https://huggingface.co/cointegrated/rubert-tiny
36. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-Agnostic BERT sentence embedding. In: ArXiv. URL: https://arxiv.org/abs/2007.01852 (2020).
37. URL: https://huggingface.co/sentence-transformers/LaBSE
38. Kuratov, Y., Arkhipov, M.: Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. In: ArXiv. URL: https://arxiv.org/abs/1905.07213 (2019).
39. URL: https://natasha.github.io/naeval
40. URL: https://natasha.github.io/
41. URL: https://pymorphy2.readthedocs.io/en/stable/
42. URL: https://wordstat.yandex.ru/
43. URL: https://translate.yandex.ru/
44. URL: https://opus.nlpl.eu/opus-100.php
45. URL: https://tatoeba.org/ru/
46. URL: https://scikit-learn.org/stable/index.html