BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **20**, No 4
Sofia • 2020 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.2478/cait-2020-0049

All-words Word Sense Disambiguation for Russian Using Automatically Generated Text Collection

Bolshina Angelina[1], *Natalia Loukachevitch*[2]

[1]Lomonosov Moscow State University, Moscow, Russia
[2]Kazan Federal University, Kazan, Russia
E-mails: angelina_ku@mail.ru louk_nat@mail.ru

***Abstract*: *The limited amount of the sense annotated data is a big challenge for the word sense disambiguation task. As a solution to this problem, we propose an algorithm of automatic generation and labelling of the training collections based on the monosemous relatives concept*. *In this article we explore the limits of this algorithm: we employ it to harvest training collections for all ambiguous nouns, verbs and adjectives presented in RuWordNet thesaurus and then evaluate the quality of the obtained collections. We demonstrate that our approach can create high-quality labelled collections with almost full-coverage of the RuWordNet polysemous words. Furthermore, we show that our method can be applied to the Word-in-Context task.*

***Keywords*: *Word Sense Disambiguation, Word-in-Context task, automatic annotation of training collections, monosemous relatives, Russian dataset, RuWordNet thesaurus*.**

1. Introduction

Knowledge acquisition bottleneck is a long-standing issue in the field of Word Sense Disambiguation (WSD) [1]. Despite the advances in the state-of-the-art neural WSD models, they still rely on the large amounts of manually sense annotated data, which require many resources for obtaining it. The all-words WSD task consists in identifying correct senses of all polysemous words in a text, given a predefined sense inventory. And even though many datasets for all-words WSD have been constructed (Senseval-2 [2]; Senseval-3 task 1 [3]; SemEval-07 task 17 [4]; SemEval-13 task 12 [5]; SemEval-15 task 13 [6]), there is no sense-tagged corpus suitable for this task in Russian.

To tackle the issue of knowledge acquisition, there have been developed various automatic or semi-automatic methods of obtaining sense-tagged corpora. In our recent works [7, 8], we describe an approach, based on the concept of monosemous relatives (related unambiguous entries), that enables to generate and label a training collection automatically (The code of our algorithm is presented here: **https://github.com/loenmac/russian_wsd_data**). In this research, we will examine the applicability of our method to the lexical sample and the all-words WSD task, and also to the Word-in-Context (WiC) task.

The contributions of the paper are as follows: first, we demonstrate that our algorithm can provide a diverse sense-annotated training collection for different parts of speech. Second, by leveraging a model trained on the automatically sense-tagged collection for three parts of speech, we construct an all-words fine-grained WSD dataset, which can be used for the evaluation of WSD systems. Moreover, we show that the proposed method can create high-quality training collections that can provide competitive performance with manually annotated data on the lexical sample WSD task. Finally, we prove that a model trained on our automatically generated training collection achieves the results in the WiC task on par with a model trained on manually annotated data.

The remainder of the article is organized as follows. In Section 2 we review the existing state-of-the-art WSD models and approaches to automatic or semi-automatic acquisition of annotated training data. In Section 3 we briefly describe our method of training collections generation. In Section 4 we present the experiments that we have conducted with the training collections created with the proposed method on the lexical sample WSD task and on the WiC task. Section 5 contains the quantitative characteristics of the monosemous relatives extracted for all the polysemous words in the RuWordNet thesaurus. In Section 6 we describe the construction of the evaluation dataset for the all-words WSD task with the help of monosemous relatives and demonstrate the results of our experiment with all-words sense prediction. In Section 7 we analyze the obtained representations from the generated training collection. Finally, Section 8 summarizes the key results of the research.

2. Related work

2.1. Word sense disambiguation systems

Despite the above-mentioned problem with the sense labelled data scarcity, currently state-of-the-art WSD systems are supervised, i.e., they utilize annotated corpora for training. The best approaches nowadays are mostly based on neural networks. The models, that are best suited for the lexical sample task, compute a probability distribution over senses only for a particular polysemous word in a sequence [9-11]. This approach is computationally easier because the set of possible senses is limited to the senses of the target word. Another type of models is designed to solve the WSD task as a sequence tagging task [12, 13]. Such networks are able to manage all the ambiguous words in a sequence leveraging the whole sense inventory, so they are suitable for the all-words WSD task. WSD systems that are based on transfer learning [14, 15] work as follows: first, a neural language model is trained; second, the context vector is extracted from this model; and, finally, another algorithm is used to predict a sense.

Contextualized embeddings, like BERT [16], ELMo [17], and context2vec [18], have also proven to be suitable for the WSD task [19-22]. Some architectures of language models were developed specifically to produce contextualized representations more suitable for the WSD task [23, 24].

Another very popular line of research is the integration of external knowledge into the WSD systems. Since the very first works in the field of WSD [25], glosses have proven to be a valuable source of information, and nowadays word definitions are incorporated in the models [26-29]. In the models [30, 31], authors integrate in the architectures the information from the lexical knowledge bases. The model described in [32] is enriched with semantic lexical features.

2.2. Methods of automatic acquiring of training collections

Most languages are limited in the amount of the labelled data necessary for training WSD models. To solve this problem, various methods of creating sense-tagged corpora have been developed. One type of such techniques exploits replacements, and does not require human resources for tagging. The most popular method is that of monosemous relatives [33]. Usually WordNet [34] is used as a source for such relatives.

Monosemous relatives are those words or collocations that are related to the target ambiguous word through some connection in WordNet, and they have only one sense, i.e. belong only to one synset. Usually, synonyms are selected as relatives but in some works hypernyms and hyponyms are chosen [35]. Some researchers replace the target word with named entities [36]. The authors of [37] substitute it with meronyms and holonyms. In the article [38] a special algorithm is created in order to select the best replacement out of all words contained within synsets of the target word and neighbouring synsets. The algorithm described in [39] to construct an annotated training set is a combination of different approaches: monosemous relatives, glosses and bootstrapping. Monosemous relatives can be also used in other tasks, for example, for finding the most frequent word senses in Russian [40]. Other methods of automatic generation of training collections for WSD exploit, for example, Wikipedia and Wiktionary [41], topic signatures [42].

In the work [14] the LSTM model is used in a semi-supervised label propagation classifier to annotate unlabeled sentences and, thus, increase the size of a training dataset. Another algorithm that is based on the label propagation is MuLaN (Multilingual Label propagatioN) [43]. In their novel label propagation method, they also utilize contextualized word embeddings, information from a knowledge base and projection of the sense tags from a high-resource language to a low-resource one.

In the method described in [44], parallel corpora are used to automatically create sense-tagged corpus “via a disambiguation pipeline that exploits the interplay between a joint multilingual disambiguation algorithm and a language-independent vector-based representation of concepts and entities”.

Train-O-Matic [45] is a language-independent algorithm for creating training data. It has three main steps: first, for every sense in an inventory, a probability distribution over all words in a vocabulary is computed. Second, each word sense is weighted for all the sentences with a target polysemous word. And, finally, a target word in a sentence is labelled with a particular sense only if this sense has the highest probability among others. OneSeC [46] is another language-independent method that utilizes Wikipedia for training samples extraction and annotation. This algorithm is based on the assumption *One Sense per Wikipedia Category*: "all the occurrences of a word across Wikipedia pages in a category share the same word meaning".

The method that we propose is based on the substitution and exploits monosemous relatives (related unambiguous entries) that can be located at relatively long distances from a target ambiguous word. The main idea of this approach is as follows: the contexts of use of the selected monosemous relatives are used as training samples for a sense of a target polysemous word. In order to extract only the most suitable samples, we implemented the procedure of ranking monosemous relatives' candidates. The details of the method will be described in the next section.

3. Monosemous relatives approach to generating training data

The underlying concept of our algorithm is a concept of monosemous relatives, that is a set of unambiguous words (or phrases) related to a particular sense of a polysemous word. Our approach for collecting a training corpus is based on the substitution: for every polysemous word we select appropriate monosemous relatives, then in a text, the occurrences of these relatives are substituted by the target polysemous word and these instances are labelled with a sense tag of a monosemous relative.

A central part of our method belongs to the candidate selection and ranking algorithm. Not all monosemous relatives are suitable as a representation of a target word sense, that is why we developed a system that assigns a weight to every candidate monosemous relative, and based on this score, we obtain a rating of all possible candidates. Moreover, this algorithm helps to verify the usage of a monosemous relative in a corpus, because some words marked as monosemous in the thesaurus may have more than one sense in a corpus.

To extract the features necessary for computing candidate weights, we utilize a semantic network, namely RuWordNet thesaurus for the Russian language [47]. It is a semantic graph with the WordNet-like structure, that consists of 111.5 thousand of words and word combinations for Russian. The nodes of the graph are represented as groups of synonyms, called synsets, and the edges are relations between these groups of words. RuWordNet is used as a source for the semantic relations (e.g., synonymy, hyponymy, etc.) between a target polysemous word and its monosemous relatives. This semantic network is exploited to compute distances between words under consideration. Also, we use the senses presented in RuWordNet as a sense inventory, that means that the WSD task, that we are dealing with, can be defined as a fine-grained one.

When constructing a training set, we take into account not only the close relations like synonymy, hypernymy and hyponymy, but also far more distant ones, for example, co-hyponymy. Our findings from the previous research [7] showed, that the inclusion of the words connected to a target ambiguous word via distant relations does not have a negative effect on the performance of the WSD model. Moreover, the experiments showed that the utilization of such distant relatives enables a wider coverage of the polysemous words from the thesaurus in a training collection. In our research, the distance between the target sense of the polysemous word and its candidate monosemous relatives can reach up to four steps in the semantic graph. Candidate monosemous relatives are unambiguous words and phrases, that can be located in up to four-step relation paths to a polysemous word and include co-hyponyms, two-step (or more) hyponyms and hypernyms, and the weights of these monosemous relatives are yet to be estimated.

Another constituent of our system is the notion of ***a synset nest***. The synset nest represents a set of words (or phrases) most closely related to a particular sense of the target word, specifically target word synonyms and all the words from directly related synsets within two steps from the target word. We use this set of words when computing a score for a candidate monosemous relative in order to identify how similar is the sense of the candidate to the sense of the target polysemous word. A fragment of the nest for the word *такса* “dachshund” is given below:

1. “*охотничий пёс, охотничья собака, пёсик, четвероногий друг, псина, собака, терьер, собачонка, борзая собака*…” / “hunting dog, hunting dog, doggie, four-legged friend, dog, dog, terrier, dog, greyhound dog…”

In order to ensure, that the samples with monosemous relatives extracted from a corpus will serve as a good representation of the target sense, we employ in our candidate selection and ranking algorithm a custom word2vec embedding model trained on the same corpus from which the contexts are retrieved.

All the principal features, that we have described above, are reflected in the following formula for calculating the weight of the monosemous candidate [7]:

(1)
$$\text{Weight}_{r_j} = \sum_{k=1}^{N_k} \max[\cos(\cos(r_j, w^j_{k_1}), ..., \cos(r_j, w^j_{k_i})].$$

In this formula: r_j is the candidate monosemous relative to a sense j of the target polysemous word; N_k is the number of synsets in a synset nest; the words $w^j_{k_1}, \dots, w^j_{k_i}$ lie at the intersection of the words from the synset nest and 100 words most similar to r_j according to the word2vec model. The formula was designed to assign higher scores to those candidates, that resemble a greater number of synsets from the nest close to the target sense of the ambiguous target word.

For example, these are the monosemous relatives’ ratings for the two senses of the word *абрикос* “apricot” (relatives weights are given in brackets):

2. “Tree”: *яблоня* “apple tree” (6.3), *яблонька* “small apple tree” (4.9), *олива* “olive tree” (4.8), *смоковница* “fig tree” (3.3), *терновник* “blackthorn” (3.0), *плодовое дерево* “fruit tree” (2.9), …, etc.

3. “Fruit”: *инжир* “fig” (6.8), *яблоко* “apple” (6.4), *смоква* “fig” (6.0), *ранет* “variety of small apples” (5.7), *антоновка* “variety of apples” (4.9), *фрукт* “fruit” (4.3), …, etc.

These examples demonstrate that different sets of monosemous relatives can help to distinguish between the senses of a target polysemous word. The scores assigned to the monosemous relatives are not absolute, the range of the score values usually depends on the number of the monosemous candidates. For example, the word *лицо* ‘person’ has around 2000 candidate monosemous relatives and the highest score among them is 24, the word *идея* “concept” has eight candidates with 2.3 being the highest score, and the word *рулет* “meatloaf” has only one monosemous relative and its weight is 0.5.

4. Monosemous relatives approach in the lexical sample task

4.1. The lexical sample WSD task

For evaluation of our algorithm of generating training data in a lexical sample task, we conducted several experiments to determine the performance of the WSD models trained on our automatically generated collections and on the manually labelled one.

For evaluation of our algorithm of training data generation, we used three distinct RUSSE’18 datasets for Russian [48]. These datasets were created for the shared task on word sense induction for the Russian language. All the polysemous words are nouns. From these datasets we have taken only those words and senses which have one-to-one correspondence with the senses in RuWordNet. The final list of the target ambiguous words contains 30 words in total, each having two different senses. We call the resulting test dataset RUSSE-RuWordNet because it is a projection of RUSSE’18 sense inventory on the RuWordNet data.

We utilized two corpora for the extraction of the training samples. A news corpus consists of news articles harvested from various news sources. Another corpus is Proza.ru, a segment of Taiga corpus [49], which is compiled of works of prose fiction.

As for the approach to a collection generation, the training examples for the target ambiguous words were collected with the help of all respective unambiguous relatives with non-zero weight. The number of extracted contexts per a monosemous candidate is in direct proportion to its weight. This training collection was called a balanced one.

In order to evaluate the training collections, we applied kNN classifier to the contextualized word embeddings extracted for the target polysemous words. We exploited two distinct ELMo models – the one trained by DeepPavlov and the other by RusVectōrēs [50]. The difference between these two models is that from the first model we extracted a vector for a whole sentence with a target word, whereas from the second model we extracted a single vector for a target ambiguous word. As for BERT, we used two models: BERT-base-multilingual-cased released by Google Research and RuBERT by DeepPavlov. To extract BERT contextual representations, we concatenated the token representations from the top four hidden layers of the pre-trained transformer.

The algorithm based on the ELMo pre-trained embeddings by RusVectōrēs outperformed all other models in all the settings. The second-best model in the WSD task is RuBERT by DeepPavlov, followed by ELMo model by DeepPavlov. The lowest F1 score belongs to Multilingual BERT. The best result of WSD was obtained with RusVectōrēs ELMo model trained on the Proza.ru collection and amounted to 0.857 F1 score.

We also evaluated Proza.ru training collection applied to a more sophisticated algorithm. We implemented a bidirectional LSTM sequence labelling architecture for WSD (with two hidden layers) [12] and trained this model using the same training collection and with the custom word2vec embeddings obtained from Proza.ru corpus. The performance on the test set equals to 0.95 F1 score. Thus, we see that a more powerful algorithm trained on our data can give even better performance.

Table 1. F1 scores for ELMo- and BERT-based WSD models, balanced collections

Model	ELMo RusVectōrēs (target word)		ELMo DeepPavlov (whole sentence)		RuBERT DeepPavlov		Multilingual BERT	
kNN	Proza.ru	News collection	Proza.ru	News collection	Proza.ru	News collection	Proza.ru	News collection
k=7	**0.857**	0.815	**0.793**	**0.759**	0.802	0.768	0.723	0.683
k=9	0.856	**0.821**	0.791	0.753	**0.812**	**0.774**	**0.729**	**0.688**
biLSTM	**0.95**	-	-		-		-	

Moreover, we compared the WSD model performance trained on the automatically and manually labelled data. In this case we also used RusVectōrēs ELMo contextualized embeddings. We took the RUSSE-RuWordNet dataset; for each target sense we generated five random divisions of its samples into train and test sets in the ratio 2:1. Then we used this data to train and test five different WSD models. Among all the results obtained by each classifier, we took the maximum value, and the final performance score was the average of these five F1 values. The F1 in this setup amounted to 0.917.

Then we computed F1 score on these five test sets using our model trained on the news corpus. We obtained F1 score equal to 0.84. And, finally, we combined our news training collection with each train set described above, and measured the performance on the corresponding test sets. The F1 score was 0.94. Our results show that manually labelled data combined with the generated one can enhance the overall performance.

All these findings show us that our method of generation and labelling of training collections enables to create qualitative data sufficient to train various types of WSD algorithms.

4.2. Word-in-context (WiC) task

In this subsection we present the quantitative evaluations of our automatically generated text collection used as a training data for a model for the Word-in-Context (WiC) task.

The Word-in-Context (WiC) task was recently introduced in [51]. This task resembles WSD, however, it was designed as a binary classification task. Each instance in WiC consists of two contexts with a target word, each representing a specific sense of a polysemous word under consideration. The aim of this task is to identify whether the two contexts correspond to the same sense or not.

The main aim of our experiment is to demonstrate that our automatically generated and labelled collection can be used as a training set for a WiC model. As an evaluation dataset in this task, we have taken a training dataset from the WiC task of Russian SuperGLUE benchmark (**https://russiansuperglue.com/**). The benchmark for the WiC task consists of the three datasets: train, validation and test. The test set in the benchmark does not contain the gold keys, because it is used to only for models' predictions, that will further be automatically checked through a submission to a system. So, for our research purposes, we take the annotated train dataset. It should also be pointed out, that the sense inventory used in this dataset is different from the one we use in this study. That is why we manually aligned one sense inventory with the other. As a result, not all the polysemous words and their senses from the original dataset were included in the final evaluation dataset. So, in total, the WiC evaluation dataset consists of 161 polysemous word and 7006 context pairs. We compiled a train collection for this task using our approach presented before: the train collection contained 9105 annotated samples.

As a classifier for this task, we used a simple MLP classifier. We extracted ELMo contextualized word embeddings for the target words in given context pairs and for the target word without any context, and used these representations as an input to the classifier. The classifier trained on our generated training collection achieves 0.91 accuracy. In order to compare the WiC models performance trained on the automatically and manually annotated data, we used the training data from the benchmark as a training set for the classifier. We performed 5-fold cross-validation over this data, and the resulting classification quality is the average of 5 accuracy scores produced by these classifiers. The obtained performance equals to 0.8.

The results of WiC experiment provide evidence that our method of generating and labelling of a training collection is suitable not only for the lexical sample WSD task, but also for the WiC task.

5. Quantitative characteristics of the polysemous words in RuWordNet thesaurus

To construct training collections for the all-word WSD task, we had to extract all the candidate monosemous relatives for each polysemous word and rank them.

As an evaluation dataset, we have chosen a corpus that consists of news from the Wikinews economics section. We have already mentioned, that in the monosemous relatives selection and ranking algorithm, it is recommended to use the word2vec model trained on the same corpus that will be used for extraction of samples with monosemous relatives. Also, experiments from the research [8] have shown that “similar genres of train and test collections give higher results in the WSD task”. Due to these factors, in this research, we will exploit the news corpus as a reference corpus. This corpus contains news articles harvested from various news sources [7, 8]. The word2vec embedding model employed in the algorithm was also trained on the news corpus.

Table 2. Quantitative characteristics of the polysemous words in RuWordNet

Number of senses of a polysemous word	Number of nouns	Number of verbs	Number of adjectives
2 senses	4273	3334	1932
3 senses	997	1118	354
4 senses	399	532	83
5 senses	149	216	18
> 5 senses	76	124	5

RuWordNet contains synsets for three parts of speech: nouns (29,297 synsets), verbs (12,865 synsets) and adjectives (7636 synsets). We have applied the selection and ranking algorithm to all the polysemous words presented in the semantic graph. In the tables below, we present quantitative characteristics of the polysemous words and their monosemous relatives.

Table 3. Quantitative characteristics of the polysemous words and their monosemous relatives

Characteristics of words	Nouns	Verbs	Adjectives
Total number of monosemous words in RuWordNet	63,014	21,051	12,566
Total number of polysemous words in RuWordNet	5894	5324	2392
Total number of unique senses that polysemous words have	10,779	4916	4453
Number of "polysemous word:sense" pairs	14,358	14,048	5379
Number of "polysemous word:sense" pairs that do not have monosemous relatives	676	255	688
Number of polysemous words that have at least two senses with monosemous relatives	5511	5220	1910
Number of polysemous words that have monosemous relatives for all of their senses	5265	5080	1813
Number of unique monosemous relatives	17,173	7224	6391
Mean number of monosemous relatives per sense	36	42	78
The median of the number of monosemous relatives per sense	10	16	10

Majority of the polysemous words in the thesaurus have 2 or 3 senses, and at the same time, our method can provide 80-90% of polysemous words in each part of speech with monosemous relatives for at least two senses. More than 75% of the ambiguous words in each part of the speech have monosemous relatives for all the senses. Therefore, our algorithm enables to generate labelled training samples for most of the polysemous words in RuWordNet.

In comparison with verbs and adjectives, nouns have more unique monosemous relatives found for their target senses, which means that nouns require larger reference corpus for extraction of training samples that will be able to cover most of them. The median of monosemous relatives per sense is almost the same for different parts of speech and equals to 10 and 16, the mean number of monosemous relatives is also rather high, suggesting that usually a polysemous word is represented by several monosemous relatives. Thus, the training collection generated in such a way is very diverse as it consists of a wide variety of training samples with different monosemous relatives.

Tables 4 and 5 present the characteristics of the monosemous relatives themselves with regard to the part of the speech they were selected for.

Table 4. Characteristics of the relations between a target sense and a monosemous relative

Relation	Nouns	Verbs	Adjectives
Synonyms	2%	3%	1%
Hyponyms	13%	13%	3%
Hypernyms	5%	9%	2%
Cohyponyms	29%	31%	34%
Cohyponyms situated at three-step path	31%	27%	36%
Cohyponyms situated at four-step path	18%	14%	23%
Other	2%	3%	1%

Table 5. Distances between target polysemous senses and their monosemous relatives

Distance	Nouns	Verbs	Adjectives
0 (synset)	2%	3%	1%
1	5%	9%	2%
2	36%	40%	37%
3	36%	31%	36%
4	21%	17%	24%

The analysis of the data shows, that most of the monosemous relatives selected by our algorithm are situated on the longer distances from the target senses. It may be observed, that the proportions of the close relatives like hyponyms and hypernyms is higher for nouns and verbs, whereas for adjectives these proportions are relatively low. Moreover, such relations as cohyponyms and cohyponyms situated at three-and four-step path contribute greatly to the wide coverage of the training collection. And these facts hold true for all parts of speech under consideration. The data once again confirms our assumption that the usage of the monosemous relatives connected to the target sense with distant relations is beneficial for the automatically generated and labelled training collection.

Once we have selected monosemous relatives for all the polysemous words in RuWordNet, we can create the training collections for the all-words WSD task. This procedure and the process of creating the evaluation dataset for all-words WSD task will be described in the next section.

6. Creating training and evaluation collections for the all-words WSD task

The outline of our experiment is as follows: first, we create a training collection for the all-words WSD task, second, we train the model on this collection, and, finally, using this model we make preliminary predictions on an evaluation dataset that will be further manually checked and corrected.

As a result of the selection and ranking procedure, we obtained a ranked list of the monosemous relatives, from which we excluded all relatives with a zero weight. In this research, we take into consideration only those polysemous words that have monosemous relatives for all their senses. Also, in our study, we will employ the balanced approach to the training collection creation. This method was first exploited in [7], and the model trained on the collection harvested in this way showed better performance in comparison with the other method of compiling training collection. According to the proposed approach, all the selected monosemous relatives are used in the collection generation, but at the same time, the number of extracted contexts per a monosemous relative is in direct proportion to its weight.

For every polysemous word sense, from the news corpus, we extracted and automatically labelled around 30 examples with the method described above. It is not a sufficient amount of data for the models based on neural networks, but in the current experiment, we will use previously described nearest neighbour classification (kNN) based on the ELMo contextualized word embeddings. This algorithm has already been employed in the lexical sample WSD, and has shown good quality of predictions (0.857 F1). Moreover, it is not so data-hungry as deep neural networks.

In this work, we utilized lemmatized ELMo model by RusVectōrēs [50] trained on Taiga Corpus [49]. ELMo model for the current experiment was also chosen based on the findings of the previous studies in lexical sample WSD: it showed the best results in all the settings. For every sense-labelled sample in the training collection, we extracted a single vector for a target ambiguous word from this language model.

We have already mentioned before, that as an evaluation dataset in the current study, we take news articles from the Wikinews economics section. As a preprocessing step, we lemmatized the whole text and removed the stop words. In total this dataset consists of 107 sentences, 1777 lemmas and 1047 unique lemmas. Table 6 summarizes the information about polysemous words in this dataset.

Table 6. Characteristics of the polysemous words presented in the evaluation dataset

Characteristics of the polysemous words	Nouns	Verbs	Adjectives
Number of unique polysemous words	224	159	84
Number of all polysemous words	400	208	149

Having made predictions with kNN classifier applied to the contextualized word embeddings extracted for target polysemous words, we manually verified the predictions. Below we present the results of our investigations:

Table 7. Characteristics of the polysemous words presented in the evaluation dataset

Characteristics of the polysemous words	Nouns	Verbs	Adjectives
F1 score of kNN classifier (number of neighbours = 5)	0.8	0.72	0.8
Number of unique synsets	243	156	93
Number of words that do not have a sense in RuWordNet	11	7	1
Number of words that are a part of a collocation	14	6	1

Despite the fact that the quality of the predictions is not perfect, some preliminary sense tagging facilitated our efforts to label evaluation dataset. We were able to ascertain through our own experience that the sense annotation is very time consuming and requires much effort. There were controversial cases in which it was not obvious what sense label to choose. For example, the sense of the noun *история* "history" in the phrase *переписать историю* "rewrite the history" should be chosen from the following senses: "science", "the course of development" and "historical development". Our group decision was the last sense from the given, although it was not so obvious from the first sight. Another example relates to the verb *встретить* "to meet" in the phrase *встретить понимание* "to meet with understanding"; here we have chosen the sense "to meet in life, activity". This decision was not straightforward, we had to analyze all other senses and their examples of use, and by process of elimination, we arrived at the conclusion.

Moreover, we encountered some cases when a word has a sense that is not included in the sense inventory. Sometimes it was due to the fact that this sense is relatively recent: e. g., the word *канал* "channel" in the sense "YouTube channel". In some cases, a necessary sense is simply absent: for example, the sense "to leave without help, support" for the verb *бросить* "throw, leave" in contexts like *бросить на растерзание* "throw (someone) to the wolves". The statistics of such cases is presented in Table 6.

Apart from that, there were cases when a polysemous word was a part of the collocation, so it does not require disambiguation. Among these cases we can mention the following: the adjective *большой* “large” in the collocation *по большому счету* “to a large extent”; the verb *отдавать* “to give” in the phrase *отдавать себе отчет* “to be aware”; the noun *свет* “light” in the phrase *выйти в свет* “go out”. The number of such cases is given in the table above.

Furthermore, we found cases when part-of-speech ambiguity interfered with lexical ambiguity. The word *стали* can be either the polysemous verb “to become” (Past form, Plur.) or the monosemous noun “steel” (Sg., Gen). We perform disambiguation on the lemmatized texts, and during lemmatization this verb was incorrectly lemmatized as the noun and its lemma became *сталь* (noun) instead of *стать* (verb), so we were not able to obtain predictions for it. This problem can be partly solved by another lemmatization tool or by applying disambiguation model to a raw text. However, even in a raw text, there can be cases with the part-of-speech ambiguity.

Finally, there was one case when a polysemous word was a part of a proper noun so there was no need in disambiguation: the noun *новости* “news” in a name of the news agency *РИА Новости* “RIA Novosti”.

As for the application of the model to the labelling, we also wanted to explore the ways to reduce the amount of human intervention even more. For this we performed a simple estimation. In our experiments we employed a rather simple classification algorithm, that is easily interpretable. The number of neighbors that we utilized equals to 5. We verified the predictions of the verb senses that were made when 4 out of 5 instances of a particular sense were the closest to a target word. We found out that in 80% of these cases the sense label predicted by the model was correct. We assume, that a model with more complex architecture may give better probabilistic estimation of a predicted sense. And this can be used to filter the training samples and leave only the most probable labelled instances. Thus, such probabilistic annotation can further alleviate the manual tagging of evaluation datasets.

The main aim of this article was not to achieve high performance in the all-words WSD task, but to assess the possibility to apply our algorithm to generate all-words training collection. However, we showed that the model trained on only a few labelled samples per sense is able to attain good results of the disambiguation. It is possible that training a more sophisticated model on a greater amount of labelled data will give higher results.

The experiment, which we described in this section, demonstrated that the training collection compiled with the help of our selection and ranking algorithm is suitable for both the lexical sample WSD task, which we described in Section 4, and the all-words task. In this study, the evaluation dataset was created semi-automatically, i.e., we relied in this process on the WSD model predictions, that were then manually curated. Using this evaluation dataset, we carried out the evaluation of the all-words model trained on our generated collection. The attained performance for all parts of speech is rather good, which means that our approach is suitable for the generation of training collections for all polysemous words irrespective of their part of the speech.

7. Visualizations of the contextualized representations from the training collection

Finally, we move our focus to explore the representations obtained from the instances in our training collection and compare them to the ones extracted from the manually labelled evaluation set. The contextualized representations were also encoded by RusVectōrēs ELMo model.

Visualizations of the polysemous word representations derived from the train collection have demonstrated, that most of the senses form easily separable sense clusters. Figs 1 and 2 show the contextualized representations obtained for the two senses of the nouns *акция* “action/share” and *крона* “krona, currency/top of a tree”.

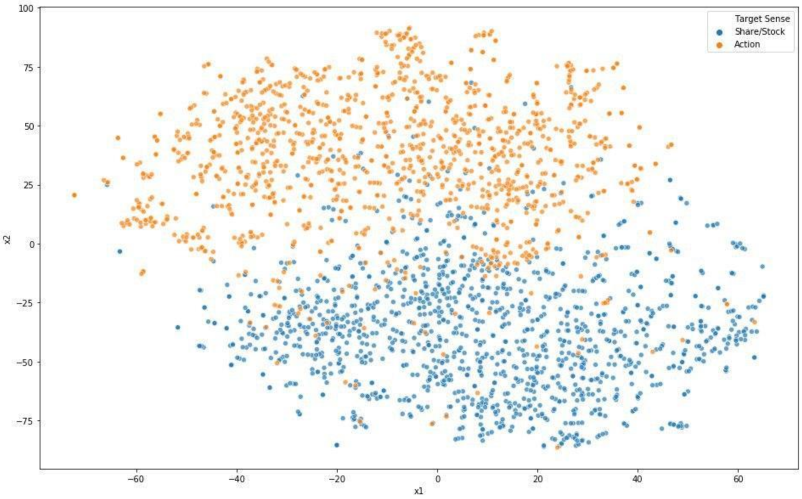

Fig. 1. Representations for the word *акция* encoded by RusVectōrēs ELMo model, contexts are taken from the automatically generated train collection; visualized with t-SNE

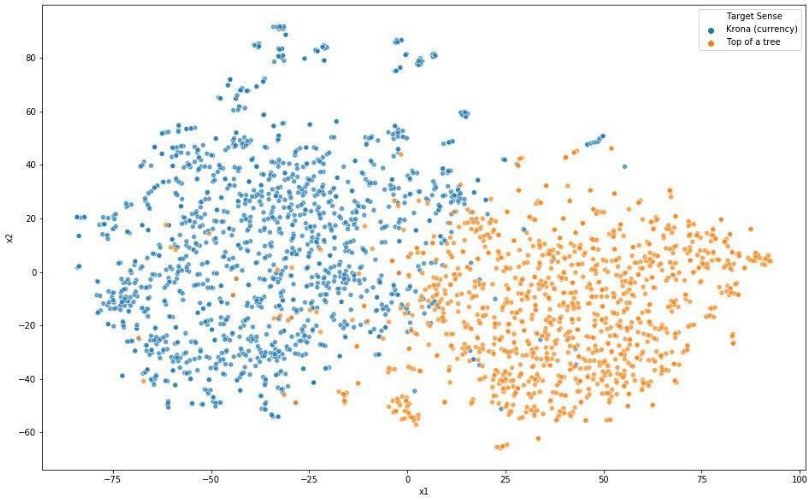

Fig. 2. Representations for the word *крона* encoded by RusVectōrēs ELMo model, contexts are taken from the automatically generated train collection; visualized with t-SNE

However, some words do not have such clear-cut sense groupings. For example, the senses of the noun *гвоздика* “clove/carnation” are shuffled and do not have any explicit “border” between them.

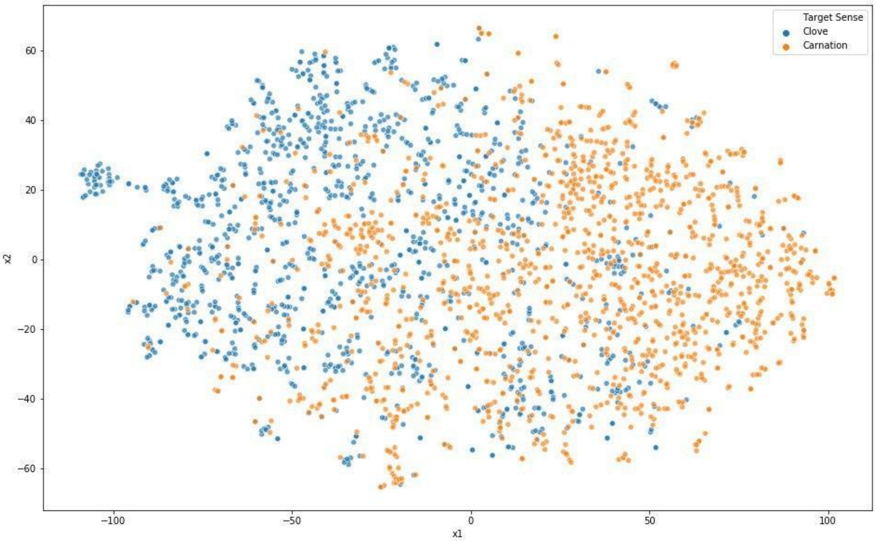

Fig. 3. Representations for the word *гвоздика* encoded by RusVectōrēs ELMo model, contexts are taken from the automatically generated train collection; visualized with t-SNE

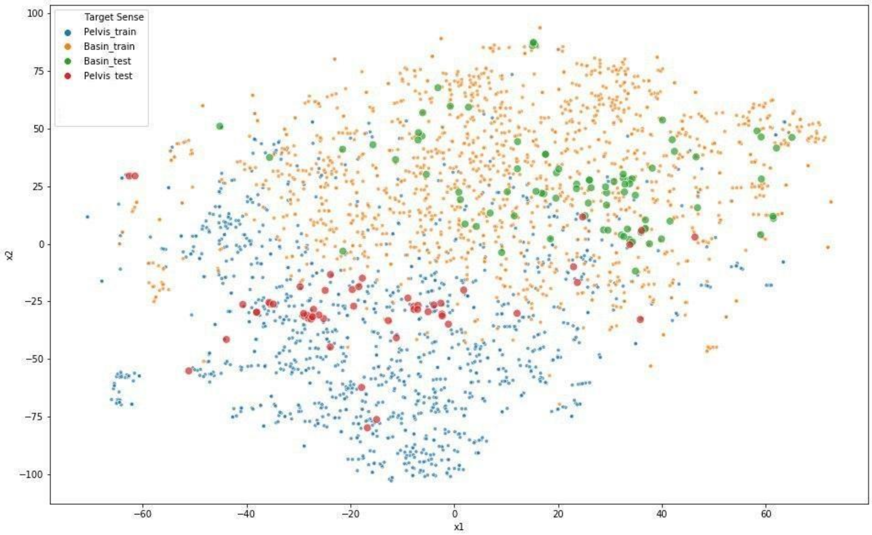

Fig. 4. Representations for the word *таз* encoded by RusVectōrēs ELMo model, senses marked with “_train” label are taken from the automatically generated train collection, senses marked with “_test” are taken from the manually annotated evaluation collection; visualized with t-SNE

Comparison of the representations extracted from the automatically generated training collection and from the manually annotated evaluation data showed that in some cases the distributions of senses were different, i.e., sense groupings occupied distinct parts of the vector space. By contrast, there are also cases when the sense clusters from the train collection resembled the ones from the evaluation set. Fig. 4 demonstrates representations for the word *таз* “pelvis/basin” in its two senses taken from the train collection (marked as “_train”) and from the evaluation dataset (marked as “_test”).

8. Conclusion

In this research, we explored the applicability of our method for the automatic generation of sense-annotated training collections to a variety of lexical-semantic tasks. We demonstrated that the proposed approach can be successfully applied to the lexical sample and the all-words WSD tasks, and also to the WiC task.

In order to assess the quality of our automatically generated collection for the all-words WSD task, we semi-automatically annotated the evaluation dataset. The performance of the all-words WSD model attests the good quality of our training collections for nouns, verbs and adjectives, which means that our approach is scalable to multiple parts of speech. Moreover, we showed that the utilization of the model pre-trained on this collection reduces the amount of human intervention during manual sense annotation of the dataset. Our experiments on the lexical sample WSD task show that the performance of the model trained on the generated data is comparable to the quality of the model trained on the manually labelled dataset.

Furthermore, we proved that our approach can provide high-quality training collections not only for the WSD task, but also for the WiC task. The model trained on the automatically generated collection outperformed the model trained on the manually labelled data in the WiC task.

***Acknowledgments*:** The work of Natalia Loukachevitch in the current study concerns formulation of the disambiguation approach for RuWordNet data, calculation of paths between synsets, criteria for selecting contexts; this work is supported by the Russian Science Foundation Grant No 19-71-10056 financed through Kazan Federal University.

References

1. Navigli, R. Word Sense Disambiguation: A Survey. – In: ACM Computing Surveys (CSUR), Vol. **41**, 2009, No 2, 10.
2. Edmonds, P., S. Cotton. Senseval-2: Overview. – In: Proc. of 2nd International Workshop on Evaluating Word Sense Disambiguation Systems, Toulouse, France, 2001, pp. 1-6.
3. Snyder, B., M. Palmer. The English All-Words Task. – In: Proc. of 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text (SENSEVAL-3), Barcelona, Spain, 2004, pp. 41-43.
4. Pradhan, S., E. Loper, D. Dligach, M. Palmer. SemEval-2007 Task-17: English Lexical Sample, SRL and All Words. – In: Proc. of SemEval, 2007, pp. 87-92.
5. Navigli, R., D. Jurgens, D. Vannella. SemEval-2013 Task 12: Multilingual Word Sense Disambiguation. – In: Proc. of SemEval 2013, 2013, pp. 222-231.
6. Moro, A., R. Navigli. Semeval2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. – In: Proc. of SemEval-2015, 2015, pp. 288-297.
7. Bolshina, A., N. Loukachevitch. Generating Training Data for Word Sense Disambiguation in Russian. – In: Proc. of Conference on Computational Linguistics and Intellectual Technologies Dialog-2020, 2020, pp. 119-132.

8. Bolshina, A., N. Loukachevitch. Comparison of Genres in Word Sense Disambiguation Using Automatically Generated Text Collections. – In: Proc. of 4th International Conference Computational Linguistics in Bulgaria (CLIB'20), 2020, pp.155-164.
9. Iacobacci, I., M. T. Pilehvar, R. Navigli. Embeddings for Word Sense Disambiguation: An Evaluation Study. – In: Proc. of ACL, Berlin, Germany, 2016, pp. 897-907.
10. Kageback, M., H. Salomonsson. Word Sense Disambiguation Using a Bidirectional LSTM. – In: Proc. of COLING, 2016, pp. 51-56.
11. Uslu, T., A. Mehler, D. Baumartz, A. Henlein, W. Hemati. FastSense: An Efficient Word Sense Disambiguation Classifier. – In: Proc. of LREC, 2018, pp. 1042-1046.
12. Raganato, A., C. D. Bovi, R. Navigli. Neural Sequence Learning Models for Word Sense Disambiguation. – In: Proc. of EMNLP, 2017, pp. 1156-1167.
13. Vial, L., B. Lecouteux, D. Schwab. Improving the Coverage and the Generalization Ability of Neural Word Sense Disambiguation through Hypernymy and Hyponymy Relationships. – arXiv preprint arXiv:1811.00960, 2018.
14. Yuan, D., J. Richardson, R. Doherty, C. Evans, E. Altendorf. Semi-Supervised Word Sense Disambiguation with Neural Models. – In: Proc. of COLING, 2016, pp. 1374-1385.
15. Le, M., M. Postma, J. Urbani, P. Vossen. A Deep Dive into Word Sense Disambiguation with LSTM. – In: Proc. of COLING, Association for Computational Linguistics, 2018, pp. 354-365.
16. Devlin, J., M.-W. Chang, K. Lee, K. Toutanova. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. – In: Proc. of 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2019, pp. 4171-4186.
17. Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, L. Zettlemoyer. Deep Contextualized Word Representations. – In: Proc. of 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 2227-2237.
18. Melamud, O., J. Goldberger, I. Dagan. Context2vec: Learning Generic Context Embedding with Bidirectional LSTM. – In: Proc. of COLING, 2016, pp. 51-61.
19. Kutuzov, A., E. Kuzmenko. To Lemmatize or Not to Lemmatize: How Word Normalisation Affects ELMo Performance in Word Sense Disambiguation. – In: Proc. of 1st NLPL Workshop on Deep Learning for Natural Language Processing, 2019, pp. 22-28.
20. Wiedemann, G., S. Remus, A. Chawla, C. Biemann. Does BERT Make Any Sense? Interpretable Word Sense Disambiguation with Contextualized Embeddings. – arXiv preprint arXiv:1909.10430, 2019.
21. Du, J., F. Qi, M. Sun. Using BERT for Word Sense Disambiguation. – In: arXiv preprint arXiv:1909.08358, 2019.
22. Hadiwinoto, C., H. T. Ng, W. C. Gan. Improved Word Sense Disambiguation Using Pre-Trained Contextualized Word Representations. – In: arXiv preprint arXiv:1910.00194, 2019.
23. Bevilacqua, M., R. Navigli. Quasi Bidirectional Encoder Representations from Transformers for Word Sense Disambiguation. – In: Proc. of International Conference on Recent Advances in Natural Language Processing (RANLP'19), 2019, pp. 122-131.
24. Levine, Y., B. Lenz, O. Dagan, D. Padnos, O. Sharir, S. Shalev-Shwartz, Y. Shoham. Sensebert: Driving Some Sense into BERT. – arXiv preprint arXiv:1908.05646, 2019.
25. Lesk, M. Automatic Sense Disambiguation Using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. – In: Proc. of International Conference on Systems Documentation, 1986.
26. Luo, F., T. Liu, Q. Xia, B. Chang, Z. Sui. Incorporating Glosses into Neural Word Sense Disambiguation. – arXiv preprint arXiv:1805.08028, 2018.
27. Huang, L., C. Sun, X. Qiu, X. Huang. GlossBERT: BERT for Word Sense Disambiguation with Gloss Knowledge. – arXiv preprint arXiv:1908.07245, 2019.
28. Blevins, T., L. Zettlemoyer. Moving Down the Long Tail of Word Sense Disambiguation with Gloss-Informed Biencoders. – arXiv preprint arXiv:2005.02590, 2020.

29. Loureiro, D., A. Jorge. Language Modelling Makes Sense: Propagating Representations through Wordnet for Full-Coverage Word Sense Disambiguation. – In: arXiv preprint arXiv:1906.10007, 2019.
30. Kumar, S., S. Jat, K. Saxena, P. Talukdar. Zero-Shot Word Sense Disambiguation Using Sense Definition Embeddings. – In: Proc. of 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5670-5681.
31. Bevilacqua, M., R. Navigli. Breaking through the 80% Glass Ceiling: Raising the State of the Art in Word Sense Disambiguation by Incorporating Knowledge Graph Information. – In: Proc. of 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 2854-2864.
32. Melacci, S., A. Globo, L. Rigutini. Enhancing Modern Supervised Word Sense Disambiguation Models by Semantic Lexical Resources. – In: Proc. of Eleventh International Conference on Language Resources and Evaluation (LREC'18), 2018.
33. Leacock, C., G. A. Miller, M. Chodorow. Using Corpus Statistics and WordNet Relations for Sense Identification. – Computational Linguistics, Vol. **24**, 1998, No 1, pp. 147-165.
34. Miller, G. WordNet: A Lexical Database for English. – Communications of the ACM, Vol. **38**, 1995, No 11, pp. 39-41.
35. Przybyła, P. How Big is Big Enough? Unsupervised Word Sense Disambiguation Using a Very Large Corpus. – arXiv preprint arXiv:1710.07960, 2017.
36. Mihalcea, R., D. I. Moldovan. An Iterative Approach to Word Sense Disambiguation. – In: FLAIRS Conference, 2000, pp. 219-223.
37. Seo, H. C., H. Chung, H. C. Rim, S. H. Myaeng, S. H. Kim. Unsupervised Word Sense Disambiguation Using WordNet Relatives. – Computer Speech & Language SPEC. – ISS, Vol. **18**, 2004, No 3, pp. 253-273.
38. Yuret, D. KU: Word Sense Disambiguation by Substitution. – In: Proc. of 4th International Workshop on Semantic Evaluations, Association for Computational Linguistics, 2007, pp. 207-213.
39. Mihalcea, R. Bootstrapping Large Sense Tagged Corpora. – In: Proc. of 3rd International Conference on Language Resources and Evaluation (LREC'02). Vol. **1999**. Las Palmas, Canary Islands, Spain, 2002.
40. Loukachevitch, N., I. Chetviorkin. Determining the Most Frequent Senses Using Russian Linguistic Ontology RuThes. – In: Proc. of Workshop on Semantic Resources and Semantic Annotation for Natural Language Processing and the Digital Humanities at NODALIDA 2015, 2015, pp. 21-27.
41. Henrich, V., E. Hinrichs, T. Vodolazova. Webcage: A Web Harvested Corpus Annotated with GermaNet Senses. – In: Proc. of 13th Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, 2012, pp. 387-396.
42. Agirre, E., O. L. De Lacalle. Publicly Available Topic Signatures for All WordNet Nominal Senses. – In: Proc: of LREC, 2004.
43. Barba, E., L. Procopio, N. Campolungo, T. Pasini, R. Navigli. MuLaN: Multilingual Label PropagatioN for Word Sense Disambiguation. – In: Proc. of IJCAI, 2020.
44. Bovi, C. D., J. Camacho-Collados, A. Raganato, R. Navigli. Eurosense: Automatic Harvesting of Multilingual Sense Annotations from Parallel Text. – In: Proc. of 55th Annual Meeting of the Association for Computational Linguistics, Vol. **2**, 2017, pp. 594-600.
45. Pasini, T., R. Navigli. Train-o-Matic: Large-Scale Supervised Word Sense Disambiguation in Multiple Languages without Manual Training Data. – In: Proc. of 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp.78-88.
46. Scarlini, B., T. Pasini, R. Navigli. Just "OneSeC" for Producing Multilingual Sense-Annotated Data. – In: Proc. of 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 699-709.
47. Loukachevitch, N. V., G. Lashevich, A. A. Gerasimova, V. V. Ivanov, B. V. Dobrov. Creating Russian WordNet by Conversion. – In: Proc. of Conference on Computational Linguistics and Intellectual Technologies Dialog-2016, 2016, pp. 405-415.

48. Panchenko, A., A. Lopukhina, D. Ustalov, K. Lopukhin, N. Arefyev, A. Leontyev, N. Loukachevitch. RUSSE'2018: A Shared Task on Word Sense Induction for the Russian Language. – In: Proc. of Computational Linguistics and Intellectual Technologies: Papers from the Annual International Conference "Dialogue", Moscow, Russia. RSUH, 2018, pp. 547-564.

49. Shavrina, T., O. Shapovalova. To the Methodology of Corpus Construction for Machine Learning: "Taiga" Syntax Tree Corpus and Parser. – In: Proc. of "CORPORA2017", International Conference, Saint-Petersburg, 2017.

50. Kutuzov, A., E. Kuzmenko. WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models. – In: D. Ignatov et al., Eds. Analysis of Images, Social Networks and Texts, AIST 2016. Communications in Computer and Information Science. Vol. **661**. Cham, Springer, 2017.

51. Pilehvar, M. T, J. Camacho-Collados. WiC: The Word-in-Context Dataset for Evaluating Context-Sensitive Meaning Representations. – In: Proc. of NAACL-HLT, 2019, pp. 1267-1273.

Received: 15.10.2020; Accepted: 29.10.2020 (fast track)