Word Sense Induction for Russian Texts Using BERT

Aleksandr Slapoguzov, Konstantin Malyuga, Evgenij Tsopa
ITMO University
St.Petersburg, Russia
slapoguzov@gmail.com, konstantin.malyuga@gmail.com, evgenij.tsopa@cs.ifmo.ru

***Abstract*—This article considers an unsupervised approach called word sense induction for resolving word sense disambiguation in the natural languages. The resolution of word sense disambiguation is one of the most important tasks in the natural text processing area, as it is the key problem of many other tasks in this field. Clustering of vector word representations was used to resolve sense ambiguity. Word translation into the vector representation was done with the RuBERT language model which was initialized with BERT and pre-trained on the Russian part of Wikipedia and news data language model. The Affinity Propagation algorithm was applied for clustering in this work. The main feature of this algorithm is not to require the number of clusters as an input parameter. Using this algorithm along with the BERT model led us to the resulting score 0.81 ARI [1] that is comparable to other methods and can be used to resolve the word sense disambiguation. The results of this work could be used in such areas as information search, information extracting, and different tasks connected with semantic networks.**

I. Introduction

With the increasing amount of generated information, the task of natural language text processing becomes more and more relevant. One of the distinguishing features of such texts is the presence of ambiguities in them when words or some language constructions can be interpreted in different ways. Word Sense Disambiguation(WSD) - is an area in natural language processing that deals with such problem. Its main task is to choose the sense and meaning of a polysemantic word in a certain context. There are generally three approaches in Word Sense Disambiguation:

1) Based on knowledge base;
2) Based on supervised learning;
3) Based on unsupervised learning.

The approach based on unsupervised learning is also called Word Sense Induction(WSI). Its difference is that the context with words, not just the specific meaning of the word, is determined. Such contexts are divided into clusters where each word is used in the same sense. Consider the following examples with the word "rock":

1) My favourite rock band is AC/DC;
2) There are a lot of picturesque rocks in the Altai region;
3) The oldest rock on Earth is more than 4 billion years old.

[1] Adjusted Random Index(ARI) shows a similarity between two data clusters (see Formula 1 on page 3)

The result of WSI work will be the clusterization of the second and the third sentences in a one group, and the first sentence - in another group. The WSI approach is usually used in cases where it is not possible to create a complete knowledge base for the analyzed domain or with lack of annotated texts for using approaches based on supervised learning. Both of these problems are present in the Russian language, therefore, the WSI research is of particular interest in this case.

In this research, we consider applying the BERT(Bidirectional Encoder Representations from Transformers) [1] language representation model to extracting "word embeddings" from Russian texts and their clustering using the Affinity Propagation algorithm. Word embedding is a method of representing words in text using vectors. Each word from a text is presented like a vector of real numbers. Words with the same meaning have similar vectors.

BERT is a pre-trained language model on a large text corpus (like Wikipedia) using the following approach: 15% of input words are masked with a special token ([MASK]) and then the model is training to predict only masked words.This approach let to extract context-aware word representations.

The rest of the article has the following structure:

- Section II - Related works - describes relevant researches in the field of Word Sense Induction;
- Section III - Dataset structure - contains a description of the datasets from the RUSSE'18 shared task;
- Section IV - Our approach - provides the detailed algorithm of the employed approach;
- Section V - Results - consists of research results with graphs and comparative tables;
- Section VI - Conclusion - describes conclusions about the further use of BERT in WSI tasks.

II. Related works

Word sense disambiguation(WSD) problem is one of the most challenging and oldest problems in Natural Language Processing. Initially, this problem was considered [2] as a task of Machine Translation. Researchers used knowledge bases, statistical information about words and contexts to solve this task. Soon it was understood that solving the WSD problem using a computer requires modeling of all world knowledge, and therefore it is impossible. Unfortunately, it is still true and there is no algorithm that is able to resolve ambiguity

with 100% accuracy. However, significant progress has been achieved since 1960, and modern approaches show 70-80% accuracy.

In the 70s, researchers faced a lack of large amounts of machine-readable knowledge when trying to solve WSD using AI methods. By the 1980s, the first large-scale lexical resources appeared (e.g. Oxford Advanced Learner's Dictionary of Current English), which made it possible to automate methods of extracting knowledge, but these methods still depended on the amount of knowledge available. Online dictionary WordNet [3] and statistical methodologies brought a revolution in WSD task in the 90s. The problem of resolving lexical ambiguity became a problem to which all possible supervised machine learning techniques are applicable.

A. Approaches for the English language

The current state-of-the-art result for word sense disambiguation task has been reached by Michele Bevilacqua and Roberto Navigli [4]. They developed a new neural WSD architecture that uses WordNet graph and pretrained synset embeddings. Such approach allows them to extract embeddings and relational information, thus decreasing knowledge acquisition bottleneck. Their solution reached 80% F1 and this is the best result for now. The main problem that has been solved during their research is predicting meanings which were not found in the training set. Another way to do it is to use WSI that doesn't require any annotated datasets and lexical databases.

WSI task was actively researched during several Semeval contests for the English language. For a long time the best results were based on sophisticated graphical models [4] [5] [6] but recent work from Amrami and Goldberg [7] showed better scores(FNMI 52.1, FBC 84.7, AVG 66.4). They decided to change the method of calculating the number of clusters: instead of a fixed amount of clusters they tried using a dynamic number of clusters. Also, they found the way of analyzing the resulting sense cluster - to consider prominent word substitutes. This solution led to better scores in comparison with sophisticated graphical models. As a result they developed Language-model Substitution with Dynamic Patterns and this model does clustering of lexical substitutes derived from BERT deep masked LM [1]. Amrami and Goldberg noted that using BERT allows them to reach a very significant improvement in WSI scores.

B. Approaches for the Russian language

A similar WSI contest called RUSSE'18 was arranged for the Russian language [8]. Overall 18 teams participated in it and the best approach used a pre-trained Continuous Bag of Words Model(CBOW). The list of the nearest neighbours was used for clustering. As a result, they scored 0.52 ARI [9] points (average across 3 datasets).

Also, there was another interesting work [10] during RUSSE'18 that is close to this research. This work is based on clustering weighted average of word embeddings for each context. RusVectōrēs [11] models were used for getting word embeddings and Affinity Propagation was used as a clustering algorithm. This approach showed 0.71 ARI score and was ranked at the second position for wiki-wiki dataset. RusVectōrēs contains several models but the best result was achieved using the ruscorpora_upos_skipgram_300_5_2018 model that was trained on the Russian National Corpus (RNC) [12] using the Continuous Skip-gram algorithm [13].

There is one more research on the task of grouping occurrences of an ambiguous word according to their meaning by N. Arefyev, B. Sheludko and A. Panchenko [14]: the main idea of this solution is to use the left and right contexts and words with similar meanings together. This approach leads to a better result than in the approach proposed by Amrami and Goldberg that used neural bidirectional language model and symmetric patterns for word sense induction task [15].

III. Dataset structure

The key part of the resolving word sense disambiguation problem is datasets preparation. For this research it was decided to use three datasets from the RUSSE'18 shared task with both test and training parts:

1) wiki-wiki. The dataset is based in Russian wikipedia. To prepare this dataset, sense sets were taken from the article titles and the articles themselves were considered contexts;
2) bts-rnc: The dataset is based on the sense inventory of the Large Explanatory Dictionary of Russian (Bolshoj Tolkovyj Slovar', BTS) and Russian National Corpus was used for contexts;
3) active-dict: The dataset is based on the Active Dictionary of Russian (Aktivnyj slovar' russkogo jazyka). Word senses were taken as the sense inventory from this dictionary, explanations and examples with that words were used as contexts.

TABLE I. The dataset statistics

Dataset	Inventory	Corpus	Split	# of words	# of senses	Avg. # of senses	# of contexts
wiki-wiki	Wikipedia	Wikipedia	train	4	8	2.0	439
wiki-wiki	Wikipedia	Wikipedia	test	5	12	2.4	539
bts-rnc	BTS	RNC	train	30	96	3.2	3 491
bts-rnc	BTS	RNC	test	51	153	3.0	6 556
active-dict	Active Dict.	Active Dict.	train	85	312	3.7	2 073
active-dict	Active Dict.	Active Dict.	test	168	555	3.3	3 729

These datasets have different granularity of their sense inventories and the text corpora from which the contexts were taken. Together they complement each other.

Each dataset consists of two parts: train and test datasets. For train set there were from 4 to 85 words with ambiguous sense (amount of words: wiki-wiki - 4, bts-rnc - 30, active-dict - 85) and hundreds and even thousands contexts for each ambiguous word (amount of contexts: wiki-wiki - 439, bts-rnc - 3491, active-dict - 2073). More detailed statistics about datasets are shown in Table I. The train dataset has the number of senses for each word. As opposed to the training dataset the test set does not have annotation with sense. Thus, the idea is to find all the senses for the words from test datasets using all the contexts provided.

The authors of [8] decided to evaluate the system performance for each dataset separately. This solution is reasonable because of the specificity of the wiki-wiki dataset from the two others: it has a more stable sense structure than bts-rnc and active-dict datasets - the average amount of word meanings in active-dict is significantly higher than in wiki-wiki (2.0 vs 3.7).

Also, it is important to notice that the type of senses in these datasets is different: the wiki-wiki set mostly consists of homonyms - words that have different meanings but are pronounced the same or spelled the same - and usually such words have different usage areas. As for bts-rnc and active-dict, they have lots of examples with related words with different senses due to metonymy and other semantic shifts. So there are two different problems: to find the correct meaning for homonyms and to reveal the exact meaning for polysemous words. This circumstance, perhaps, explains a such big difference between the results with wiki-wiki dataset and bts-rnc and active-dict datasets.

IV. Our approach

Even simple WSI approaches are able to reach good results and it was shown in Kutuzov's research [10]. Indeed, his approach contains only two steps:

1) Convert contexts to fixed-length vectors that manifest their semantics. RusVectōrēs models were used for this purpose;
2) Apply Affinity Propagation [16] algorithm to cluster word vectors into groups that represent word senses.

Such solution makes good predictions (ranked 2nd, 0.71 ARI score) for wiki-wiki dataset but works worse for bts-rnc (0.24 ARI) and active-dict (0.21 ARI). At the same time Amrami and Goldberg [7] said that using the BERT model for converting words to vectors improved WSI scores significantly. Based on this statement, it is reasonable to check the ability of BERT to improve the WSI scores for the Russian language. Therefore our approach is similar to Kutuzov's one but RusVectōrēs was replaced with the BERT model.

BERT is a language representation model that was trained to predict words and it is the main reason why BERT might be a good source for word embeddings. The original BERT model was trained from multilingual unlabeled texts. However, there are other BERT models that were additionally pre-trained for a specific language or domain. RuBERT [17] is one of such models that was trained on the Russian part of Wikipedia and news data. In this research we use Sentence RuBERT that was initialized with RuBERT and fine-tuned on SNLI [18] google-translated to russian and on russian part of XNLI dev set [19].

Context vectors may be built in several ways. For example, it can be built from word embeddings of one word or sum of all word embeddings in a context. In the Kutuzov's work "semantic fingerprint" [20] approach was applied for this. "Semantic fingerprint" is an averaged vector of unique word embeddings in a context. In this work, a similar technique was used. Its difference is that punctuation marks and prepositions are filtered out first.

Affinity Propagation algorithm is similar to k-medoids because it finds "exemplars" that are representative of clusters but the difference is that it doesn't require the number of clusters as an input parameter. However, the algorithm is tuned using damping and preference parameters. The first parameter is used for exponential smoothing and affects the convergence of the algorithm. The second parameter (preference) adds noise to the similarity matrix thus it affects a number of clusters. Words usually have between 1 and 5 senses therefore the preference parameter should have low values to reduce the number of clusters.

In this research, we use the first 4 layers of the BERT language model as word embeddings. Each layer has 768 parameters as a result our word vector has 3072 values. Sometimes such high-dimensional data might be clustered poorly [21]. Dimensionality reduction algorithms are applying

$$\overbrace{ARI}^{\text{Adjusted Index}} = \frac{\overbrace{\sum_{ij}\binom{n_{ij}}{2}}^{\text{Index}} - \overbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}^{\text{Expected Index}}}{\underbrace{\frac{1}{2}[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}]}_{\text{Max Index}} - \underbrace{[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}]/\binom{n}{2}}_{\text{Expected Index}}}$$

Formula 1. ARI score, where n_{ij}, a_i, b_i are values from the contingency table and n_{ij} denotes the number of objects in common between i and j clusters.

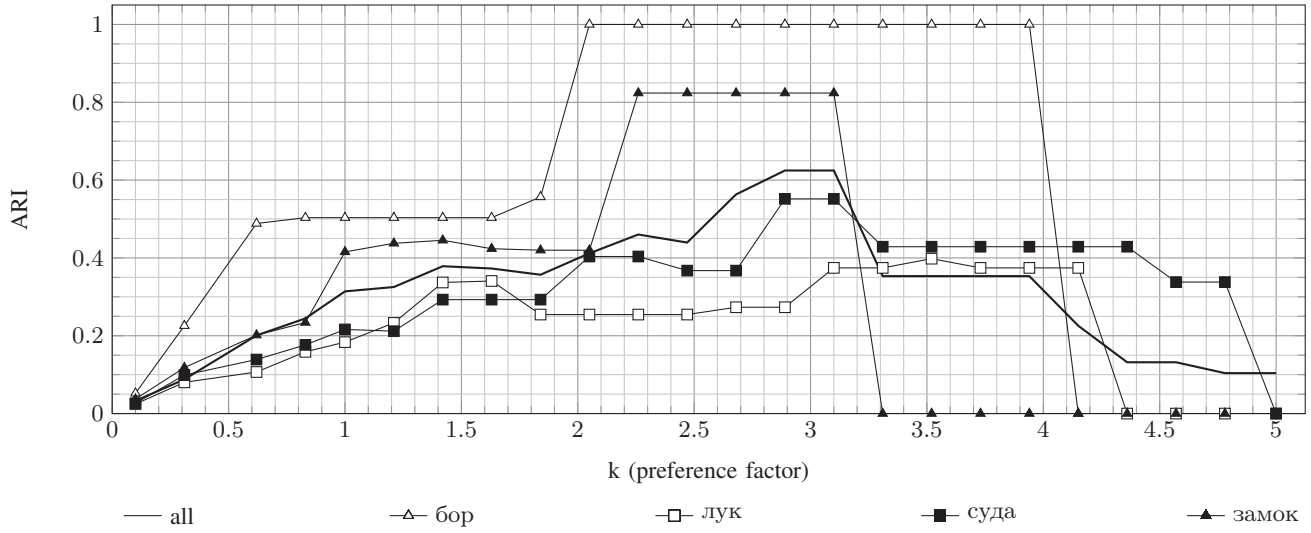

Fig. 1. The results of tuning Affinity Propagation algorithm for RUSSE'18 training datasets

in such cases. For this purpose, we use Principal Components Analysis that is able to reduce the characteristics that are less influential on the data.

Adjusted Random Index(ARI) [9] was used to measure the performance of clustering. The index shows a similarity between two data clusters and can be computed by Formula 1.

To summarize, our approach contains the following steps:

1) Get word embeddings from the BERT language model for each sentence;
2) Remove punctuation marks, prepositions and conjunctions;
3) Build "semantic fingerprint" as an averaged vector of unique word embeddings;
4) Reduce dimension of data using PCA algorithm;
5) Apply Affinity Propagation algorithm to cluster word embeddings into groups that represent word senses.

V. Results

The proposed approach was checked with 3 russian datasets from RUSSE'2018 shared task. Firstly, the Affinity Propagation algorithm was tuned using training datasets and grid search technique. The preference parameter is calculated using the following formula:

$$preference = -1 * (max\_distance * k)^2 \qquad (2)$$

where $max\_distance$ — maximum Euclidean distance between all vectors and k — input parameter that varies from 0,01 to 5. Multiplication by -1 is necessary here to decrease the similarity matrix and the number of clusters. The results of grid search for wiki-wiki training dataset is shown in Fig. 1. This dataset contains 4 words with the following meanings:

- "бор": "chemical element boron" or "pinery wood";
- "замок": "castle" or "lock";
- "лук": "onion" or "bow(weapon)";
- "суда": "ships" or "court of law".

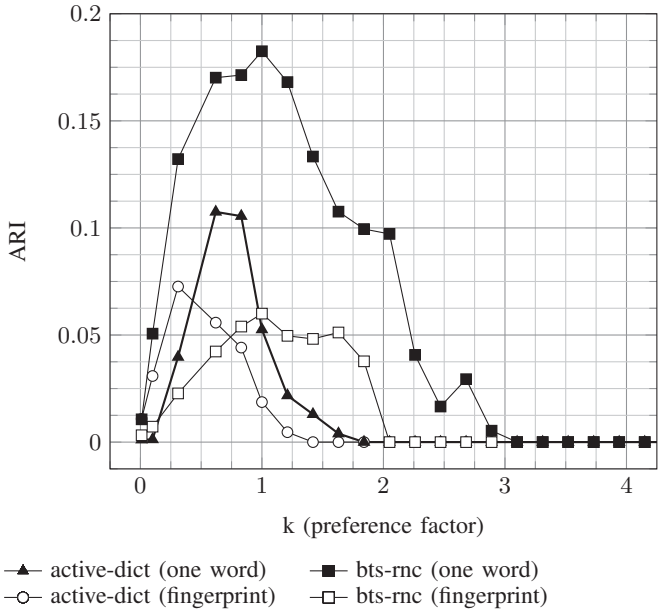

Fig. 2. The results of tuning Affinity Propagation algorithm for bts-rnc and active-dict datasets

ARI score for the wiki-wiki training dataset is reaching a maximum of 0.64 when k = 2.9 then it is decreasing sharply. It is happening because the word 'замок' that had 2 clusters and 1.0 ARI score in the previous step now has only 1 cluster that reduces its ARI score to 0. The same thing is happening with word 'бор' after k = 3.95. Such abrupt changes may signal that the model is overtrained. However, it was not proven for the test datatest.

Also, we investigated how the reduction of the data dimension using the PCA algorithm affects the results for the wiki-wiki training dataset. As shown in Fig. 3 using the PCA algorithm to reduce data dimension led to increasing ARI score

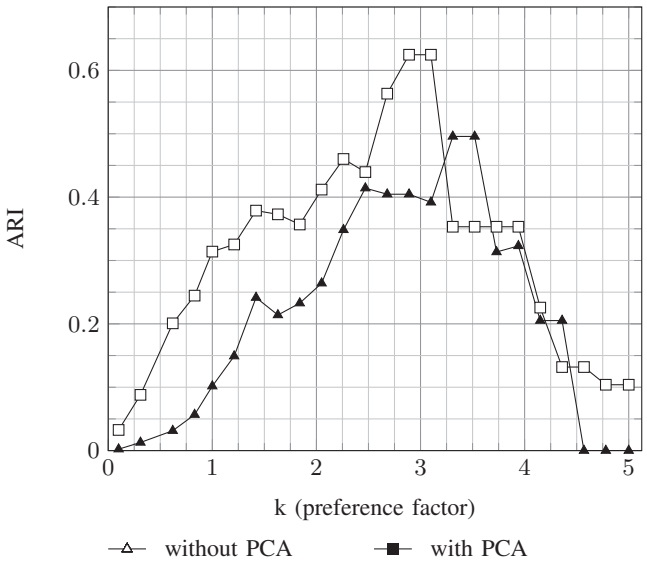

Fig. 3. Comparing approaches with and without the PCA algorithm for the wiki-wiki training datasets

from 0.50 to 0.64.

The same tuning was done for bts-rnc and active-dict datasets, the results are shown in Fig. 2. Initially, they were tuned in the same way as wiki-wiki dataset. As you can see (thin lines) in Fig. 2. it wasn't as successful as for the wiki-wiki dataset. The reason for this is different structure of the datasets: the training wiki-wiki dataset has more stable structure (two senses for each word), more examples per sense, and usually a word appears several times in an example. You can see these differences in Fig. 4 and 5.

The points in Fig. 4.2 are denser than Fig. 4.1 and there are no obvious clusters. Such situation might happen because of using "semantic fingerprint" that averages all vectors in a context. To check this statement we replaced a "semantic fingerprint" with a particular vector of a word and disabled dimension reduction. The results for this configuration are presented in Fig. 2 (bold lines) and Fig. 5. The best average ARI score increased from 0.05 to 0.19 for the bts-rnc training set and from 0.05 to 0.1 for active-dict dataset. You can also see the changes for the word "" ("environment", "social

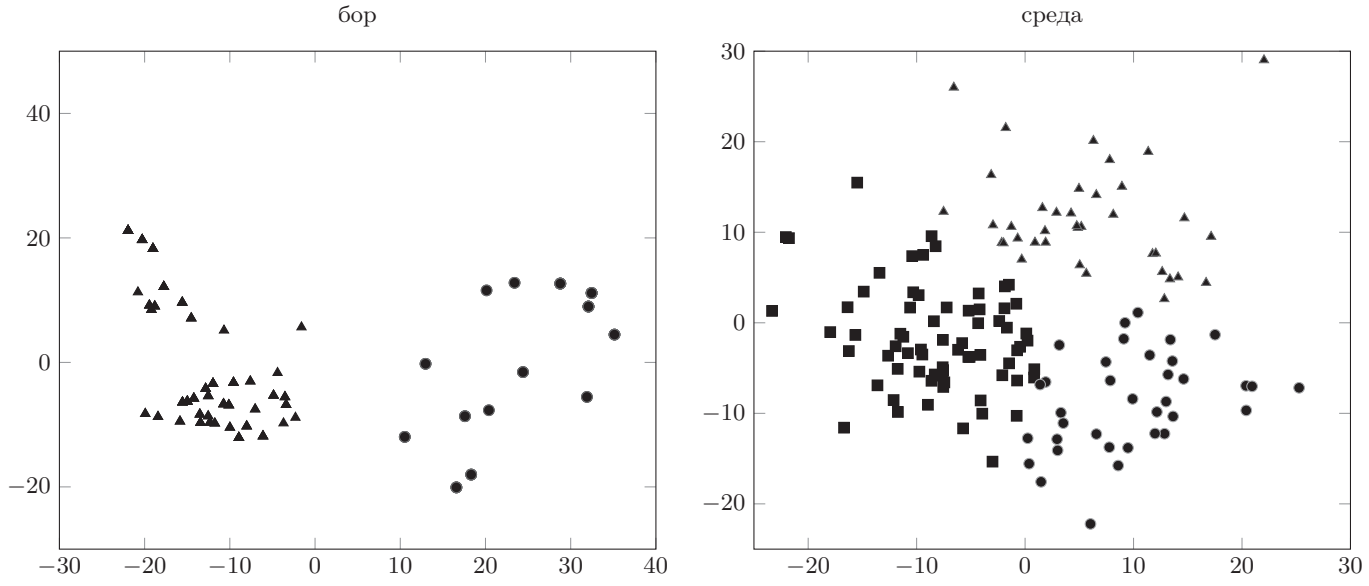

Fig. 4. (1) Clustering of the "бор" contexts ("chemical element" and "pinery") from the wiki-wiki training dataset and (2) "среда" contexts ("environment", "social context" and "Wednesday") from the bts-rnc training dataset using "semantic fingerprint"

context" and "Wednesday") in graph Fig. 5, its ARI score increased from 0.30 to 0.62.

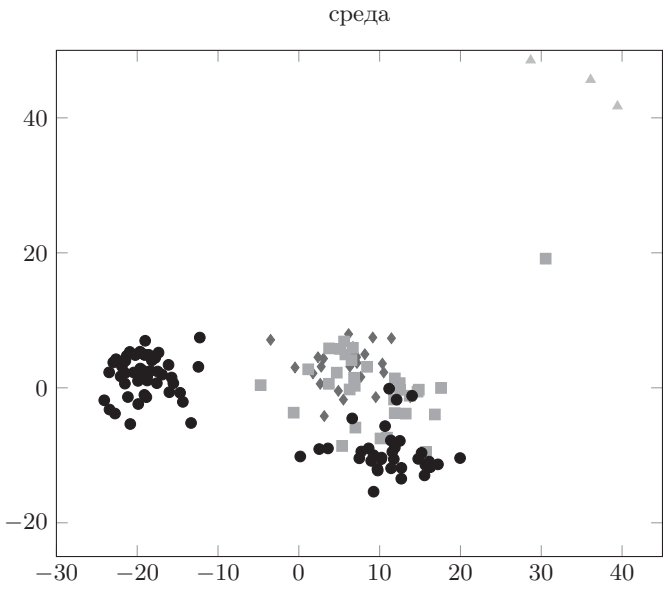

Fig. 5. Clustering of the "среда" contexts ("environment", "social context" and "Wednesday") from the bts-rnc training dataset using one word embedding. ARI = 0.62

The final results of evaluating our approach on the test sets are presented in Table II. These scores were obtained with the following parameters: k = 2.9 for the wiki-wiki test dataset, k = 1.0 for bts-rnc and k=0.6 for active-dict. It should be noted that unlike bts-rnc and active-dict datasets the wiki-wiki dataset was evaluated with "semantic fingerprint" and dimension reduction.

TABLE II. OVERALL RESULTS (EVALUATED ON THE TEST SETS)

Dataset name	ARI score of our approach	ARI score of the baseline approach [10]
wiki-wiki	0.81	0.71
bts-rnc	0.21	0.24
active-dict	0.11	0.21

Also, the ARI score for the wiki-wiki test dataset is higher than the score for the training dataset that means the model was not overtrained on the training set. As a result, our approach to the wiki-wiki dataset showed a significant improvement(from 0.71 to 0.81 ARI) in comparison with the baseline approach by Kutuzov [10]. However, the ARI scores for bts-rnc and active-dict are less than for wiki-wiki but none of the competing systems managed to achieve ARI higher than 0.34 for these datasets.

VI. CONCLUSIONS

We describe a simple and effective WSI method based on the clustering of vector word representations, where word translation into the vector representation was done with the BERT language model. The approach was tested on three datasets: wiki-wiki, bts-rnc and active-dict. We increased the result for the wiki-wiki dataset by 15%. For the other two datasets the results changed slightly. This might be caused by the difference of used datasets: wiki-wiki dataset mostly contains homonyms and the WSI task with this dataset is to find the correct sense for disambiguated words with non-related senses. Bts-rnc and active-dict datasets contain mostly related disambiguated words, and it is quite a different problem to search for the exact meaning of polysemous words, and, probably, these tasks should be investigated separately.

In contrast to Kutuzov [10] work, we used the BERT language representation model for getting word embeddings rather than the RusVectōrēs [11] models. As for other similar researches for the English language using BERT models significantly increased the results, and we have successfully applied this knowledge for the Russian language. The obtained results suggest further research on using the BERT language model in more complicated methods.

REFERENCES

[1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[2] W. N. Locke and A. D. Booth, *Machine translation of languages: fourteen essays*. Published jointly by Technology Press of the Massachusetts Institute of ..., 1955.

[3] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[4] M. Bevilacqua and R. Navigli, "Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 2854–2864.

[5] J. Wang, M. Bansal, K. Gimpel, B. D. Ziebart, and C. T. Yu, "A sense-topic model for word sense induction with unsupervised data enrichment," *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 59–71, 2015.

[6] R. K. Amplayo, S.-w. Hwang, and M. Song, "Autosense model for word sense induction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 6212–6219.

[7] A. Amrami and Y. Goldberg, "Towards better substitution-based word sense induction," *arXiv preprint arXiv:1905.12598*, 2019.

[8] A. Panchenko, A. Lopukhina, D. Ustalov, K. Lopukhin, N. Arefyev, A. Leontyev, and N. Loukachevitch, "Russe'2018: a shared task on word sense induction for the russian language," *arXiv preprint arXiv:1803.05795*, 2018.

[9] L. Hubert and P. Arabie, "Comparing partitions journal of classification 2 193–218," *Google Scholar*, 1985.

[10] A. Kutuzov, "Russian word sense induction by clustering averaged word embeddings," *arXiv preprint arXiv:1805.02258*, 2018.

[11] A. Kutuzov and E. Kuzmenko, "Webvectors: a toolkit for building web interfaces for vector semantic models," in *International Conference on Analysis of Images, Social Networks and Texts*. Springer, 2016, pp. 155–161.

[12] V. A. Plungian, "Why we make russian national corpus? [?" *Otechestvennye Zapiski*, no. 2, pp. 296–308, 2005.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] N. Arefyev, B. Sheludko, and A. Panchenko, "Combining lexical substitutes in neural word sense induction," in *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, 2019, pp. 62–70.

[15] A. Amrami and Y. Goldberg, "Word sense induction with neural bilm and symmetric patterns," *arXiv preprint arXiv:1808.08518*, 2018.

[16] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *science*, vol. 315, no. 5814, pp. 972–976, 2007.

[17] Y. Kuratov and M. Arkhipov, "Adaptation of deep bidirectional multilingual transformers for russian language," *arXiv preprint arXiv:1905.07213*, 2019.

[18] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[19] A. Conneau, G. Lample, R. Rinott, A. Williams, S. R. Bowman, H. Schwenk, and V. Stoyanov, "Xnli: Evaluating cross-lingual sentence representations," *arXiv preprint arXiv:1809.05053*, 2018.

[20] A. Kutuzov, M. Kopotev, T. Sviridenko, and L. Ivanova, "Clustering comparable corpora of russian and ukrainian academic texts: Word embeddings and semantic fingerprints," *arXiv preprint arXiv:1604.05372*, 2016.

[21] J. Simanullang, M. Zarlis, and E. M. Zamzami, "Performance improvement of clustering affinity propagation method using principal component analysis," in *Journal of Physics: Conference Series*, vol. 1566, no. 1. IOP Publishing, 2020, p. 012126.