Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference "Dialogue 2020"

Moscow, June 17–20, 2020

DISTRIBUTIONAL MODELS AND AUXILIARY METHODS FOR DETERMINING THE HYPERNYMS OF WORDS IN RUSSIAN

Yadrintsev V. V. (vvyadrincev@gmail.com)

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences; Peoples Friendship University of Russia (RUDN University), Moscow, Russia

Ryzhova A. A. (Anastasiia.Ryzhova@skoltech.ru)

Skolkovo Institute of Science and Technology, Moscow, Russia

Sochenkov I. V. (sochenkov@isa.ru)

Federal Research Center "Computer Science and Control" of the Russian Academy of Sciences, Moscow, Russia

This paper describes our participation in the first shared task on Automatic Taxonomy Construction for the Russian language RUSSE'2020. The goal of this task is the following: input words (neologisms that are not yet included in the taxonomy) need to be associated with the appropriate hypernyms from an existing taxonomy. For example, for the input word "duck", it is expected that participants will provide a list of its ten hypernyms-synsets to which the word can most likely be attributed, such as "animal," "bird" and so on. An input word can refer to one, two, or more "parents" at the same time.

In this article we are trying to answer the following question: what results can be achieved using only "raw" vectors from distributional models without additional training? The article presents the results for several pre-trained models that are based on fastText, Elmo, and BERT algorithms. Also, an out-of-vocabulary analysis was performed for the models under consideration. Taking into account all public scores from the leaderboards, we showed the results corresponding to the following places in the ranking: the 3rd place on public nouns, the 2nd on private nouns, the 4th on public verbs, and the 4th on private verbs.

Keywords: vector models, hypernym discovery, fastText, Elmo, BERT, rusvectores, RuWordNet

DOI: 10.28995/2075-7182-2020-19-762-772

ДИСТРИБУТИВНЫЕ МОДЕЛИ И ВСПОМОГАТЕЛЬНЫЕ МЕТОДЫ ДЛЯ ОПРЕДЕЛЕНИЯ ГИПЕРОНИМОВ СЛОВ РУССКОГО ЯЗЫКА

Ядринцев В. В. (vvyadrincev@gmail.com)
ФИЦ ИУ РАН; РУДН, Москва

Рыжова А. А. (Anastasiia.Ryzhova@skoltech.ru)
Сколковский институт науки и технологий, Москва

Соченков И. В. (sochenkov@isa.ru)
ФИЦ ИУ РАН, Москва

В работе описывается наше участие в первой задаче по автоматическому построению таксономии для русского языка RUSSE’2020. Цель этой задачи заключается в следующем: входным неизвестным словам (неологизмам, которых ещё нет в таксономии) нужно сопоставить гиперонимы из существующей таксономии. Например, ожидается, что для слова «утка» участники предоставят список десяти его наиболее вероятных синсетов-гиперонимов («животное», «птица» и т. д.). Входное слово может одновременно относиться к одному, двум или более «родителям». В этой статье мы показываем, каких результатов можно достичь, используя только «сырые» векторы из дистрибутивных моделей без какого-либо дополнительного обучения. В работе представлены результаты для нескольких предобученных моделей, которые основаны на алгоритмах fastText, Elmo и BERT. Кроме того, для рассматриваемых моделей был проведён анализ полноты словарей. Принимая во внимание все опубликованные результаты рейтингов участников, мы показали результаты, соответствующие следующим местам: третьему на «общедоступных» существительных, второму на «конфиденциальных» существительных, четвертому на «общедоступных» глаголах и четвертому на «конфиденциальных» глаголах.

Ключевые слова: векторные модели, определение гиперонима слова, fastText, BERT, rusvectores, RuWordNet

1. Introduction

A hypernym—hyponym relation is a word/phrase pair (x, y) such that x is a hyponym of y, the “is-a” relationship, for example, *“a dog is an animal.”* Here “dog” is a hyponym for “animal”, and “animal” is a hypernym for the word “dog”.

Identifying hypernymic relations has a lot of applications in Natural Language Processing, especially in semantically intensive tasks, such as Question Answering, Textual Entailment, and semantic search systems. These relations play a crucial role in thesauri construction, but it is challenging and not effective to extract them manually.

We participated in the shared task on Automatic Taxonomy Construction for the Russian language (RUSSE'2020[1]). The goal of this task is the following: neologisms need to be associated with the appropriate hypernyms from an existing taxonomy. As a taxonomy the RuWordNet (Russian WordNet) is used, the format of which is similar to the English WordNet format. The task consists of two subtasks:

- nouns (two test sets: public and private)
- verbs (two test sets: public and private).

The organizers provided a baseline that leverages pre-trained models to obtain word vectors. Our method is an improvement on the baseline. We intentionally employed a simple approach to identifying a hypernym of a word, which we describe below. The reason for this was that we were interested in whether the Russian taxonomy construction task can be solved using already available algorithms and pre-trained models without additional training. Even using the simple approach, we showed results that were not lower than the 4th place (from more than 13 participants) on each of the test sets.

The rest of the paper is organized as follows. **Section 2** briefly outlines the previous work related to our task. In **Section 3** we present the datasets offered by the shared task organizers and used pre-trained models. **Section 4** provides the details of the employed approach. In **Section 5** we describe the results, and in Section 6 we conclude.

2. Related work

Many automatic methods for identifying hypernymic relations have been explored in the last years. There are two popular ways of extracting such relations, a pattern-based one and a distributional one. The pattern-based approach uses the joint co-occurrence of the word and its hypernyms in texts [1], [11], while the distributional approach exploits distributional representations of words [3], [16]. Marti Hearst first introduced the now widely used pattern-based method for the English language in 1992 [5], [13]. She manually designed the patterns for hypernym—hyponym extraction from texts. For example, the pattern "such NP as NP" helps to extract such pair as "author, Shakespeare" from the sentence "such authors as Shakespeare." Shared tasks are described in the paper of the Organizers [12].

For the Russian language, this problem is not so highly investigated. In [14] Sabirova et al. propose a rule-based method for hypernym—hyponym extraction from Russian texts. They created six patterns, e.g., *"Y—вид/тип/форма/разновидность/сорт X (Y is a kind/type/form/sort of X)"*, and then applied finite-state transducers to extract the patterns from texts. In [6] the researchers clustered the definitions from the large dataset (using [7] as a starting point) and then extracted hypernym candidates using patterns for verbose candidates. As a complementary method they trained the SVM classifier to obtain the best candidates.

[1] https://russe.nlpub.org/2020/isa/

3. Data overview

RuWordNet thesaurus and train data are described in the paper of the Organizers [12]. In the present work we use the following pre-trained models:

1. ft_cc_ru_300[2],
2. RuBERT[3],
3. ruscorpora_none_fasttextskipgram_300_2_2019,
4. tayga_none_fasttextcbow_300_10_2019,
5. araneum_none_fasttextcbow_300_5_2018,
6. tayga_lemmas_elmo_2048_2019.

The first one, **ft_cc_ru_300,** includes pre-trained word vectors for Russian from *Facebook* [4]. The second one, **RuBERT,** is an adopted BERT for Russian [10]. Models 3–6 contain pre-trained word vectors for Russian from **rusvectores**[4] [9].

Please note that in the RuBert model we only consider the hidden layer with dimension 3,072, using it as word vectors. This idea is taken from the baseline provided by the organizers of the competition. Accordingly, vector dimensions of models 1, 3–5 are 300, model 2—3,072, and model 6—1,024.

It is most likely that the largest text corpus was used for **ft_cc_ru_300**, which includes *Wikipedia* and *Common Crawl*[5] (we do not know the exact volume of crawl data for Russian, but roughly 24 terabytes of plain text were used for 157 languages [4]). **RuBERT** was trained on the *Wikipedia* and *news data*, **ruscorpora_none_fasttextskipgram_300_2_2019**—on *Russian National Corpus*[6]. **Tayga_none_fasttextcbow_300_10_2019** and **tayga_lemmas_elmo_2048_2019** were trained on the *TAIGA*[7] corpus [15]. Finally, **araneum_none_fasttextcbow_300_5_2018** was obtained by training on the *Araneum Russicum Maximum* [2].

4. Our approach

The first subsection briefly describes the baseline. The following subsections describe additional steps taken to improve the baseline. Proposed improvements significantly increased the results on the test samples.

4.1. Baseline

This subsection briefly describes the baseline provided by the competition organizers. The common-crawl fasttext (300-d) model is used to obtain synset vectors

2 https://fasttext.cc/docs/en/crawl-vectors.html

3 http://docs.deeppavlov.ai/en/master/features/models/bert.html

4 https://rusvectores.org/ru/models/

5 https://commoncrawl.org/

6 http://ruscorpora.ru/

7 https://tatianashavrina.github.io/taiga_site/

and unknown word vectors. The synset vector is the average word vector of all synset senses. Variables *nouns_cnt* and *verbs_cnt* denote the number of synsets-nouns and synsets-verbs respectively. As noted earlier, the total number of nouns is ~29,300, of verbs—~7,500. For the existing taxonomy, separate vector matrices are created for nouns and verbs of sizes *nouns_cnt* × 300 and *verbs_cnt* × 300 respectively. For each unknown word, the closest synsets are searched by cosine measure, and, depending on the approach, they are considered as synonyms or hypernyms.

4.2. Proposed improvements

To achieve better results, we proposed the following improvements:

1. Addition of ranking at the final stage: sorting synsets based on the recalculated rate for each synset_id. It gave the most significant improvement in results (the MAP was increased by *5–6%*) and will be described separately in section 4.2.1.

2. Extension of the string representation of the synset. The following fields were considered as parameters: **ruthes_name**, **definition**, **sense_name**, **sense_lemma**, and **sense_main_word**. We have discovered that for nouns a combination of two fields (**ruthes_name**, **sense_name**) is better, while for verbs all fields combined work the best. The above combinations were applied for all models except **RuBERT**. For **RuBERT** we leveraged a standard string representation, consisting of the **sense_name**s of the senses. The usage of the non-standard combinations improved the results only slightly (the MAP increased by *1–3%*). Here is an example of a synset: **synset_id**=*"109649-N"* **ruthes_name**=*"ДЗЮДО"* *("judo")* **definition**=*"японская борьба, произошедшая из джиу-джитсу, олимпийский вид спорта" ("Japanese wrestling that took place from Jiu-Jitsu, an Olympic sport"*). Here are the senses of the synset *109649-N*:

- **sense_id**=*"109649-N-181880"* **sense_name**=*"БОРЬБА ДЗЮДО"* **sense_lemma**=*"БОРЬБА ДЗЮДО"* **sense_main_word**=*"БОРЬБА";*
- **sense_id**=*"109649-N-136843"* **sense_name**=*"ДЗЮДО"* **sense_lemma**=*"ДЗЮДО"* **sense_main_word**=*""*.

Thus, for the synset *109649-N* the following line will be initial: *"ДЗЮДО<sep>БОРЬБА ДЗЮДО<sep>ДЗЮДО"* (in case the fields **ruthes_name** and **name** are used). Space plays the role of the separator <sep>.

3. Addition of other relationships between synsets. We tried adding the "domain" relation. For example, word "*judo*"[8] is a part of *"sport" (спорт)* and "*amateur wrestling*" (*спортивная борьба*) domains, and *"judo"* has hypernyms "Martial Arts" *(боевые искусства)* and "east Martial Arts" (*восточные единоборства*). However, it worsened the results slightly.

4. Usage of train data to get "parents." It influenced minimal deterioration.

5. Normalization of the words of the string representation of synsets. It improved the results (the MAP was increased by *1–3%*) and will be described separately in section 4.2.2.

6. Lemmatization of all words from a string representation of a synset. The results have changed slightly.

[8] http://www.ruwordnet.ru/ru/search/%D0%94%D0%97%D0%AE%D0%94%D0%9E

4.2.1. Ranking

This improvement consists of adding parameters to the original algorithm. The ranking algorithm uses the following parameters:

- The number of synsets-associates—$\boldsymbol{k}$.
- The number of final synsets-hypernyms—$\boldsymbol{n}$.
- The probability that the synset-associate is a hypernym of the input word—$\boldsymbol{p_1}$.
- The probability that the hypernyms of the synset-associate are the input word hypernyms—$\boldsymbol{p_2}$.
- The probability that the hypernyms of the hypernyms of the synset-associate are the input word hypernyms—$\boldsymbol{p_3}$.

For the synsets a matrix of vectors $\boldsymbol{M}$ is formed. Vector $\boldsymbol{V}$ is assigned an input word. The number of rows in the matrix $\boldsymbol{M}$ is the same for all models: it is equal to the number of synsets-nouns or synsets-verbs. The number of columns, as well as the dimension of the vector $\boldsymbol{V}$, depends on the model. It is mentioned in the corresponding section 3. The relevance $\boldsymbol{R}$ is calculated using an unnormalized measure. In the beginning, each synset from the thesaurus is associated with $\boldsymbol{R = 0}$. At the first step of the algorithm, a search is performed (by cosine measure) for the $\boldsymbol{k}$ closest synsets-associates. Technically, we look for vectors that are close to $\boldsymbol{V}$ in the matrix $\boldsymbol{M}$. Assume $\boldsymbol{r}$ is a cosine measure for a synset-associate. There is a simple recalculation of $\boldsymbol{R}$, consisting of three steps:

- $\boldsymbol{R}$ of the synset-associate increases by $\boldsymbol{r \cdot p_1}$;
- $\boldsymbol{R}$ of hypernyms of the synset-associate increases by $\boldsymbol{r \cdot p_2}$;
- $\boldsymbol{R}$ of hypernyms of synsets from previous step increases by $\boldsymbol{r \cdot p_3}$.

Hypernyms in the second and third steps are taken from the thesaurus using the "*hypernym*" relation. At the end of the algorithm, the top $\boldsymbol{n}$ (by $\boldsymbol{R}$) synsets-hypernyms are selected for the answer.

4.2.2. Normalization

- Firstly, all words are converted to lowercase.
- Secondly, all punctuation except for a hyphen ("-") is replaced by a space. The list of punctuation symbols is as follows: '$', '!', '.', '?', '+', '[', '\\xa0', '%', \""\", '\u00bb', '*', ';', ':', ')', '@', '/', '\u00a7', '`', '_', '\u00b7', ',', '#', '\u2013', '\\\\', '-', '\\xad', '{', '\u2014', '>', '|', '\u00ab', ']', '}', '\"', '&', '=', '^', '<', '(', '~', '\u00b0'. Note that non-standard characters from the RuWordNet words are also included in this list.
- Then, using the pymorphy2[9] [8] morphological analyzer, functional words (prepositions, conjunctions, etc) are removed. We restricted the tags *NPRO, PRED, PREP, CONJ, PRCL, INTJ*.
- If "*Geox*" is present in the word tag list, the first letter is replaced with a large one. If parameter *lowercase* == *true*, then this change does not work.

[9] ttps://pymorphy2.readthedocs.io/

4.3. Out-of-vocabulary analysis

Table 3 presents the out-of-vocabulary analysis for all models (except **RuBERT**) on public, private, and RuWordNet words. RuWordNet words are normalized in the same way as in evaluation. The first line in Table 3 shows the number of unique words separately for nouns and verbs. It should be noted that the string representation of the synset can include nouns, verbs, and other parts of speech, regardless of the synset part of the speech. Thus, the number of words for N (53,082) in the latest column does not mean that all 53 thousand words are nouns.

It was interesting for us to see how well the words are presented in the vocabularies of models. The observations from Table 3 are the following:

- **ft_cc_ru_300** best represents the words of RuWordNet (coverage is 86.8% for Nouns and 89.2% for Verbs).
- **araneum_none_fasttextcbow_300_5_2018** best represents the test nouns (coverage is 97.1% for Public Nouns and 96.9% for Private Nouns).
- **tayga_lemmas_elmo_2048_2019** best represents the test verbs (coverage is 89.1% for Public Verbs and 88.8% for Private Verbs).

Table 3. Out-of-vocabulary analysis

Model	public N = 762 public V = 175 *in vocab (rate)* *PoS*	private N = 1525 private V = 350 *in vocab (rate)* *PoS*	RuWordNet synsets. *normalized=True,* *lemmatized=False.* N = 53,082; V = 27427
ft_cc_ru_300	722 (0.947) N 140 (0.8) V	1,443 (0.946) N 279 (0.797) V	**46,079 (0.868) N** **24,470 (0.892) V**
ruscorpora_none_fasttextskipgram_300_2_2019	548 (0.719) N 145 (0.828) V	1,094 (0.717) N 281 (0.802) V	30,625 (0.576) N 17,659 (0.643) V
tayga_none_fasttextcbow_300_10_2019	550 (0.721) N 153 (0.874) V	1,100 (0.721) N 302 (0.862) V	31,089 (0.585) N 17,975 (0.655) V
araneum_none_fasttextcbow_300_5_2018	**740 (0.971) N** 100 (0.571) V	**1,479 (0.969) N** 208 (0.594) V	31,341 (0.590) N 13,827 (0.504) V
tayga_lemmas_elmo_2048_2019	592 (0.776) N **156 (0.891) V**	1,209 (0.792) N **311 (0.888) V**	32,563 (0.613) N 18,640 (0.679) V

5. Results

The results are presented in Table 4. Note that we used RuBERT in an uncommon way. Also, we would like to highlight that in this case the set of fields for the string representation of the synset is different from other models.

Here we list the same parameters for all models in Table 4:

- The ranking algorithm is used with the parameters $\boldsymbol{p1} = 0.1$, $\boldsymbol{p2} = 1.0$, $\boldsymbol{p3}=1.0$, $\boldsymbol{k} = 10$ and $\boldsymbol{n} = 10$. These parameters were obtained with the grid search. The following values were considered: for ***p1, p2, p3***—0.1, 0.5, 1.0, 1.5; for ***k***—3, 5, 7, 10, 20, 50, 100; for ***p***—3, 5, 7, 10.
- Neologisms (input words) are lowercase.

- The comparison indicator is the MAP[10] provided by the organizers of the competition.

String representations of the synsets are different for **RuBERT**: all models except RuBERT used **ruthes_name** and **sense_name** for **Nouns** and all possible descriptions for **Verbs**. RuBERT used just **sense_name** for both **Nouns** and **Verbs**.

Next, we describe the names of the columns and rows of the tables. The first column is the name of the model. The second and the next columns are results for a **Public** or **Private** test set. "Lemmas" means that morphological analysis and lemmatization by pymorphy2 are performed. The main cells show the result, the letter after the MAP denotes part of speech (N—nouns, V—verbs).

Table 4. Results by models

Model	Public			Private		
	lowercase *MAP PoS*	lemmas *MAP PoS*	lemmas lowercase *MAP PoS*	lowercase *MAP PoS*	lemmas *MAP PoS*	lemmas lowercase *MAP PoS*
ft_cc_ru_300	0.511 N 0.291 V	**0.512 N** 0.287 V	**0.512 N** 0.286 V	0.512 N 0.359 V	**0.516 N** 0.345 V	0.515 N 0.346 V
tayga_none_fasttext cbow_300_10_2019	0.250 N 0.210 V	0.249 N 0.220 V	0.248 N 0.219 V	0.254 N 0.253 V	0.254 N 0.253 V	0.255 N 0.253 V
araneum_none_fasttext cbow_300_5_2018	0.345 N 0.188 V	0.350 N 0.209 V	0.350 N 0.208 V	0.365 N 0.235 V	0.371 N 0.229 V	0.372 N 0.229 V
tayga_lemmas_ elmo_2048_2019	0.360 N **0.334 V**	0.365 N 0.314 V	0.367 N 0.307 V	0.410 N **0.387 V**	0.405 N 0.379 V	0.405 N 0.370 V
RuBERT	0.329 N 0.183 V	—	—	0.318 N 0.190 V	—	—

Here are some observations from Table 4:

- Lemmatization (of synset representations) did not significantly affect the results. Some models showed a slightly better result, and some a little worse.
- **ft_cc_ru_300** performed the best results on nouns.
- **tayga_lemmas_elmo_2048_2019** performed the best results on verbs.
- On Private Verbs models show the results which are 4–6% better than on Public Verbs. However, we do not observe this on Nouns, except the **tayga_lemmas_elmo_2048_2019** model.
- The application of the model "**RuBERT**" in this way did not show high results.

Finally, Table 5 shows our best-submitted results compared to the baseline and the best results in the competition. As one can observe, the results we have obtained are competitive.

10 https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)#Mean_average_precision

Table 5. The best results of our experiments, which we submitted to the RUSSE shared task. **Our** team submitted results through the participant **vvyadrincev**

Dataset	Model, method	Test MAP (public)	Rank (public)	Test MAP (private)	Rank (private)
Nouns	Unknown, best in the competition	0.5590	1 of 14[11]	0.5522	1 of 17[12]
Nouns	ft_cc_ru_300, **our**	0.5115	3 of 14	0.5163	2 of 17
Nouns	ft_cc_ru_300, baseline	0.4348	9 of 14	0.4210	9 of 17
Verbs	Unknown, best in the competition	0.4033	1 of 14[13]	0.4483	1 of 14[14]
Verbs	tayga_lemmas_ elmo_2048_2019,**our**	0.3342	4 of 14	0.3874	4 of 14
Verbs	ft_cc_ru_300, baseline	0.2759	8 of 14	0.3335	6 of 14

6. Discussion and conclusion

This article is a description of our participation in the joint task RUSSE'2020 on automatic taxonomy construction for the Russian language. We intended to create a simple method based on the baseline, using pre-trained models.

Using BERT as a distribution model for obtaining vectors, we were not able to achieve high results. Therefore, as future work, we want to train RuBERT for classifying strings like "*<WORD> is a <PARENT SYNSET>*". However, we can face some challenges. Firstly, the string representation of synsets is often quite long. Secondly, the difficulties may arise in constructing high-quality training data, since the RuWordNet thesaurus, in our opinion, the latter is far from complete.

The following is the contribution we made:

- It is tested how the use of various fields from the RuWordNet affects the result. For nouns it has been shown that adding **ruthes_name** to the string representation of synsets leads to better results, while adding **definition, lemma,** and **main_word** does not improve the performance. For verbs it has been shown that adding all possible fields is the best solution.
- The ranking is added to the baseline and synsets-synonyms, and their "parents" and "grandparents" are taken into account. This improvement is beneficial since we got a list of synsets-candidates sorted by relevance.
- It is shown that even without additional training competitive results can be achieved. That is, using only pre-trained distributive models and adding a few steps to the baseline, you can get competitive results.

[11] Table "Practice (NOUNS)" is taken into account.

[12] Tables "Evaluation (NOUNS)" and "Post-Evaluation (NOUNS)" are taken into account.

[13] Table "Practice (VERBS)" is taken into account.

[14] Tables "Evaluation (VERBS)" and "Post-Evaluation (VERBS)" are taken into account.

- We showed that **ft_cc_ru_300** achieves the best result on nouns (compared to other models from our work) and **tayga_lemmas_elmo_2048_2019**—on verbs.
- Python source code is available online[15].

7. Acknowledgments

We express our gratitude to the RUSSE'2020 organizers for the chance to participate in an exciting shared task. We are grateful to the reviewers for careful reading of the manuscript and helpful remarks. The reported study was funded by RFBR according to the research projects №18-29-03187 and №18-37-20017 and with the support of the "RUDN University Program 5-100".

References

1. *Alain Auger and Caroline Barrière.* Pattern-based approaches to semantic relation extraction: A state-of-the-art. Terminology, 14:1–19, 01 2008.
2. *Benko V., Zakharov V.* (2016). Very Large Russian Corpora: New Opportunities and New Challenges. In Computational Linguistics and Intellectual Technologies, pp. 79–93.
3. *Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung-Chieh Shan.* Entailment above the word level in distributional semantics. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, pp. 23–32, Avignon, France, April 2012. Association for Computational Linguistics.
4. *Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov.* Learning word vectors for 157 languages. In Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018), 2018.
5. *Marti A. Hearst.* Automatic acquisition of hyponyms from large text corpora. In COLING 1992 Volume 2: The 15th International Conference on Computational Linguistics, 1992.
6. *Maria Karyaeva, Pavel Braslavski, and Yury Kiselev.* Extraction of hypernyms from dictionaries with a little help from word embeddings. In Alexander Panchenko, Wil M. van der Aalst, Michael Khachay, Panos M. Pardalos, Vladimir Batagelj, Natalia Loukachevitch, Goran Glavaˇs, Dmitry I. Ignatov, Sergei O. Kuznetsov, Olessia Koltsova, Irina A. Lomazova, Andrey V. Savchenko, Amedeo Napoli, and Marcello Pelillo, editors, Analysis of Images, Social Networks and Texts—7th International Conference, AIST 2018, Revised Selected Papers, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 76–87, Germany, 1 2018. Springer Verlag.
7. *Kiselev Y., Porshnev S., Mukhin M.* Method of extracting hyponym-hypernym relationships for nouns from definitions of explanatory dictionaries [metod izvlecheniya rodovidovykh otnosheniy mezhdu sushchestvitel'nymi iz opredeleniy

[15] https://github.com/vvyadrincev/taxonomy-enrichment

tolkovykh slovarey]. Software Engineering [Programmnaya inzheneriya]. 10, pp. 38–48, 2012.
8. *Korobov M.* Morphological Analyzer and Generator for Russian and Ukrainian Languages. Analysis of Images, Social Networks and Texts, pp. 320–332, 2015.
9. *Andrey Kutuzov and Elizaveta Kuzmenko.* WebVectors: A Toolkit for Building Web Interfaces for Vector Semantic Models, pp. 155–161. Springer International Publishing, Cham, 2017.
10. *Kuratov, Y., Arkhipov, M.* (2019). Adaptation of Deep Bidirectional Multilingual Transformers for Russian Language. arXiv preprint arXiv:1905.07213.
11. *Roberto Navigli and Paola Velardi.* Learning word-class lattices for definition and hypernym extraction. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, pp. 1318–1327, Uppsala, Sweden, July 2010. Association for Computational Linguistics.
12. *Irina Nikishina, Varvara Logacheva, Alexander Panchenko, and Natalia Loukachevitch.* RUSSE'2020: Findings of the First Taxonomy Enrichment Task for the Russian Language. In Proceedings of the 26th International Conference on Computational Linguistics and Intellectual Technologies (Dialogue-2020). Moscow, Russia, 2020.
13. *Stephen Roller, Douwe Kiela, and Maximilian Nickel.* Hearst patterns revisited: Automatic hypernym detection from large text corpora. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 358–363, Melbourne, Australia, July 2018. Association for Computational Linguistics.
14. *K. Sabirova and A. Lukanin.* Automatic extraction of hypernyms and hyponyms from Russian texts. CEUR Workshop Proceedings, 1197:35–40, 01 2014.
15. *Shavrina T., Shapovalova O.* To the methodology of corpus construction for machine learning: «Taiga» syntax tree corpus and parser. In Proceedings of "CORPORA2017", Saint Petersburg, pp. 78–84, 2017.
16. *Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller.* Learning to distinguish hypernyms and co-hyponyms. In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pp. 2249–2259, Dublin, Ireland, August 2014. Dublin City University and the Association for Computational Linguistics.