

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

Moscow, June 17–20, 2020

TAXONOMY ENRICHMENT FOR RUSSIAN: SYNSET CLASSIFICATION OUTPERFORMS LINEAR HYPONYM-HYPERNYM PROJECTIONS

Kunilovskaya M. (maria.kunilovskaya@wlv.ac.uk),
Kutuzov A. (andreku@ifi.uio.no), **Plum A.** (a.j.plum@wlv.ac.uk)
University of Wolverhampton, UK
University of Oslo, Norway
University of Wolverhampton, UK

We present the description of our system that was ranked third in the noun sub-track of the Taxonomy Enrichment for the Russian Language shared task offered by Dialogue Evaluation 2020. Our best-performing system appears against the backdrop of other methods and their combinations attempted, and its results argue in favour of Occam’s razor for this task. A simple supervised classifier was trained on static distributional embeddings of hyponym words as features and their numeric hypernym synset identifiers from the taxonomy as class labels. It outperformed more complicated approaches based on learning linear projections from hyponym embeddings to hypernym embeddings and returning synset identifiers for the nearest neighbours of the predicted vectors. Training specially tailored word embeddings for ruWordNet multi-word expressions proved to be one of the key factors for both approaches.

Key words: taxonomy enrichment, hypernymy relations, distributional semantics, word embeddings, projection learning, supervised machine learning

DOI: 10.28995/2075-7182-2020-19-474-484

ПОПОЛНЕНИЕ ТАКСОНОМИИ ДЛЯ РУССКОГО ЯЗЫКА: ЛИНЕЙНЫЕ ГИПО-ГИПЕРОНИМИЧЕСКИЕ ПРОЕКЦИИ ИЛИ КЛАССИФИКАТОР СИНСЕТОВ

В настоящей статье описывается способ расширения таксономии, который занял третье место в соревновании, объявленном в рамках Dialogue Evaluation 2020 (задача определения гиперонимических синсетов для существительных). Мы сравниваем наш наиболее эффективный подход с другими методами, которые были применены к решению поставленной задачи. Наши опыт и результаты свидетельствуют в пользу выбора более простого подхода, который изначально не представлялся многообещающим. Таким методом оказался классификатор, обученный на векторах гипонимов и идентификационных номерах соответствующих гиперонимических синсетов. Его результат значительно выше чем для метода, основанного на выучивании линейной трансформации вектора гипонима в вектор гиперонима с последующим поиском слов (и идентификаторов их синсетов), семантически похожих на предсказанные гиперонимы. Для обоих подходов важную роль играет наличие качественных дистрибутивных векторных репрезентаций для многословных единиц тезауруса ruWordNet.

Ключевые слова: пополнение таксономии, гипо-гиперонимические отношения, векторные репрезентации, линейная трансформация векторов, машинное обучение с учителем

1. Introduction

This paper reports our approaches to the shared task offered by the Taxonomy Enrichment for the Russian Language competition within Dialogue Evaluation 2020¹ (see the overview [12]). Taxonomy enrichment is a natural language processing (NLP) task where a system is required to add new entries to an existing lexical database (ontology). Dialogue Evaluation 2020 provided a list of noun and verb unigrams to be attached to the appropriate hypernym synsets in ruWordNet [10]. At first glance, this task is derived from the lexical-level hypernymy detection and extraction challenges, when hypernymy, or a ‘is-a’ relationship, is established between two lemmas or two senses in a WordNet-format taxonomy. However, taxonomy enrichment can be re-cast as a classification task where a learner is presented with the target hyponym words and the associated hypernym synset IDs as the class labels. This approach avoids learning the semantic relations between words *per se*, but directly links the semantics of hyponyms to their generic concepts represented by the synsets.

¹ <https://competitions.codalab.org/competitions/22168>

1.1. Task description

The participants were asked to return up to 10 hypernym synsets from the existing taxonomy (ruWordNet) for every target hyponym from the test set. The systems were primarily evaluated with Mean Average Precision (MAP), which measures the accuracy of target synset detection. This measure was tuned to take into account accuracy per connected component, i.e. it returned higher results if the predicted synsets include at least one correct answer for each specified set of hypernyms associated with each sense of a polysemous hyponym. For example, the Russian word ‘yasli’ [ясли] has two distinct senses and two respective sets of hypernyms: (1) a day nursery and (2) a trough. A successful system is expected to predict synsets from both sets. Further more, the organisers limited the set of correct answers by the first and second order hypernyms, which are treated indiscriminately for evaluation purposes.

The paper is structured as follows: [Section 2](#) provides an account of the research that motivated some of our solutions. In [Section 3](#), we introduce our methods and their settings, while in [Section 4](#) we describe the linguistic resources used to solve the task. Our experimental setup is discussed in [Section 5](#), followed by the presentation of the results in [Section 6](#).

2. Related work

Hypernymy, as a semantic relation between a specific and a generic term, underlies several interrelated NLP tasks: (1) hypernymy detection, when a provided pair of words is being classified as hypernymic or not, (2) hypernym discovery/extraction, when an input word is given a set of possible super-ordinates, (3) taxonomy induction/construction, when the required output is a hierarchy of concepts in a general or special domain and (4) taxonomy enrichment, when new items need to be integrated into an existing taxonomy. In this variety of hypernymy related tasks, the input lexical items are usually represented either by formal, morpho-syntactic and contextual properties of the queries (co-occurrence and pattern-based statistical approaches) or by word vectors within the distributional approach to capturing word semantics. Methodologically, in the former case a query’s hypernym is directly extracted from a pattern match, while the latter approach follows one of two scenarios: the task can be cast either as a classification of word pairs (possibly constructed from the entire corpus vocabulary) that outputs the probability of hypernymy relation existing between the two words or as learning a transformation matrix from a hyponym vector to a vector in the subspace of possible hypernyms. There have been attempts to benefit from a combination of patterns-based information and vector representations.

The obvious limitations of the pattern-based approaches include lack of coverage, when many terms receive no candidate hypernyms, and noisy results, when patterns return figurative or broad generalisations. To give a curious example from our experiments, ‘Самая серьезная **проблема** — **это человек**’ [‘The most serious **problem is (a) man**’], where the ‘is-a’ pattern returns ‘man’ as a hypernym for ‘problem’. Often, such methodologies use lexico-syntactic patterns similar to, or adapted from those introduced by Marti Hearst in 1992 and named after her [\[5\]](#). For example, Michael Oakes applied them to automatically extract hyponymy relations from

a corpus of an English pharmaceutical texts [13]. Panchenko et al. successfully used sub-string matching and pattern searches to extract candidate hypernyms from raw text in four languages (but not for Russian) in the SemEval 2016 task of automatic taxonomy generation (Task 13) [14]. The key to their success was the introduction of a classifier-based filter for the noise returned by the patterns. Hearst patterns for Russian were adapted by Kristina Sabirova and Artem Lukanin in 2014 in an attempt to facilitate the creation of the Russian thesauri and to improve query expansion algorithms in information retrieval systems [16].

Another approach to represent words in the hypernymy-related tasks is to use word embeddings. However, based on the results of the SemEval-2016: Task 14 challenge, word embeddings are not very effective, if the hypernymy detection task is formulated as a classification task. In the SemEval-2016: Task 14 setting the participants were provided with an out-of-taxonomy word sense and its definition and were expected to demonstrate high integration accuracy as to (i) the identification of the exact sense of the existing taxonomy entry that is hypernymic to the query, and (ii) the choice of the correct integration operation (add a new lemma to the existing synset or introduce a new concept). The best performing system cast the problem as a pairwise classification task to detect the correct hypernym for each hyponym [17]. Their classifier demonstrated only very moderate improvement after the feature set constrained to the formal and grammatical properties extracted from WordNet was extended to include the distributional semantics features. Overall, it is the only system (out of 13) that is slightly above the naive First Word, First Sense baseline, which picked the first word that matched the query part of speech (PoS) in the provided definition and attached the query to the first sense of it [6].

Instead of classification, Ustalov et al. implemented an approach that learns “a hypernym embedding on the basis of a hyponym embedding” [19]. It is known as projection learning and was initially applied in bilingual tasks like word translation. Ustalov et al. combined the projection learning system applied to hypernymy extraction from [4] and the negative sampling regularisation idea from [20]. They presented a distributional hypernym extraction system equipped with clustering and two options for negative sampling: their algorithm is either penalised for predicting vectors similar to the vector of the *input hyponym* (asymmetric regularisation that uses inverted hypernymy relations) or for predicting vectors similar to the vectors of the *input hyponym synonyms* (neighbour regularisation).

In essence, the task at hand is more in line with the hypernymy modelling as offered at SemEval-2018: Task 9 [3], which asked participants to retrieve suitable hypernyms from a target corpus (in our case, from ruWordNet) given an input term. The pairwise classification setup would require to evaluate all possible pairs of the ruWordNet senses in the respective PoS category for every test hyponym. This was the reason we did not use the pairwise classification approach, but went for projection learning and another kind of classification setup described below.

Note that a major practical drawback of using pre-trained word embeddings in any NLP task, including taxonomy enrichment, is the issue of out-of-vocabulary words (OOV). As we show in the next sections, this is particularly important in the context of the current shared task: more than half of noun lemmas in ruWordNet are

actually multi-word expressions (MWE), which often lack representations in the pre-trained embeddings. [Subsection 3.2](#) describes how we handled this problem.

Now we move on to explain the approaches we devised based on this analysis in more detail, including most of the ideas discussed above.

3. Methodology

3.1. Baseline approach

All our approaches are based on distributional Continuous Skipgram or Continuous Bag-of-Words embeddings [11]. For the baseline system, we represent all pairs of hyponym-hypernym words from the provided training data with the available pre-trained vectors, discarding all items not found in the embeddings vocabulary and ignoring connected components. Then we learn the optimal linear transformation of hyponym vectors into hypernym vectors and predict a hypernym vector for each new test word. We return the synset ids of k ruWordNet words most similar (based on vector cosine similarity) to the predicted vector. Duplicate ids in the list were removed. For the test words not found in the embedding model vocabulary, we return ids of the synsets that are most frequent as hypernyms (but not domains) in ruWordNet. As an alternative strategy for OOV words handling, we tried using **fastText** vectors [2], but they yielded lower results. We comment on the performance of a number of available pre-trained word embedding models using this setup in [Section 4](#).

3.2. Improvements

Some improvement was gained when we enhanced our baseline projection learner with the negative sampling regularisation as suggested in [19]. Similar to their findings, we noticed better performance for the synonym based approach, when the model is penalised for predicting hypernym vectors which are close to ‘the semantic neighbourhood of the hyponym’.

An additional increase in performance was seen after we introduced the refinement of the projection learning results using hypernyms extracted with Hearst patterns. The list of the top k words that were found to be most similar to the predicted hypernym vector was reordered to give priority to the hypernyms matched by seven Hearst patterns, inspired by Sabirova and Lukanin [16]. We implemented this filter only for nouns, but the MAP results with the filter were consistently higher for all our settings and test sets.

The obvious limitation of most of the available pre-trained word embeddings is that many multi-word entities (MWEs) are missing from their vocabularies (if any MWEs are present there at all). MWEs account for about 57.4% of the ruWordNet training data word list for nouns and for 46% for verbs. Moreover, the inability to represent MWE rendered almost 19% of noun synsets entirely inaccessible for us at the prediction stage, because they do not have any one-word lemmas (‘senses’) attached to them. For example, it was impossible for us to predict ‘butterfly’ [ба́ттерфляй]

as ‘a swimming stroke’ [стиль плавания]. We were sceptical of representing MWEs as averaged vectors of their components and decided in favour of tokenising them in a large corpus and training distributional embeddings from scratch on this tailored corpus. As expected, this yielded good results and increased the performance of the baseline system.

3.3. Classifier as an Alternative

The real boost in performance was achieved with the approach that did not initially look promising because of its simplicity. The error analysis for the previous methods demonstrated that though we were returning reasonably good results in terms of hypernym words, it was tricky to output the correct synset ids. For example, even if we predicted ‘bird’ [птица] for ‘lapping’ [чибис] and ‘art’ [искусство] for ‘folk-rhyme’ [частушка], our system did not score because the gold answers for these test words include only (5681-N, 7396-N), verbalised as ‘wild bird’ and ‘bird of passage’ and (108434-N", 7372-N), verbalised as ‘vocal (art) piece’ and ‘song, songlet’ respectively. To account for this, our simple classifier approach (see more on it in [Section 5](#)) **avoids learning relations between words** in the hyponym-hypernym pairs entirely. Instead, it directly uses hypernym synsets (or, rather, their ids) as classes of hyponym words (represented with their embeddings). At test time, the trained classifier was given a vector representation of a query word, and predicted a probability distribution over all classes (hypernym synsets from ruWordNet) for this vector. Our best submission includes 10 synset ids with the highest probabilities per test hyponym.

3.4. Failed Attempts

We attempted a number of other strategies to improve the results. This subsection contains a brief account of what did not work for this task.

As an approximate for Hearst patterns statistics, we used the corpus co-occurrence counts to refine our raw output for nouns and move the frequently co-occurring hypernyms up the list. To that end, we concatenated the four Russian corpora that were at our disposal (see Section 4 for details) and produced a frequency dictionary for sentence-based co-occurrences of the test words with each ruWordNet noun. If a lemma was seen in the top n predicted hypernyms and also in the top k co-occurring items for the test word, we moved it up the prediction list. The gains from this laborious effort did not live up to our expectations: The improvements were only marginal.

Our initial attempt to avoid the intermediary step of extracting hypernym words consisted of producing synset vectors as average vectors of all lemmas attached to the synset (‘de-worded approach’). The idea was that it would allow us to directly link hyponym words to the synset ids. However, the results for this setting were even lower than the baseline.

Finally, we unsuccessfully tried to reduce the number of same-level concepts predicted as hypernyms by weeding out the direct hyponyms of the higher level concepts in the output, hoping to move the true hypernyms up the list of predictions. The attempts to disambiguate hypernyms in the output and return only relevant ids instead

of all of them (based on measuring the similarity between the predicted vector and the averaged synset id vector) did not work either: not the least because this improvement, even if successful, affected only 30% of the output.

For brevity, we report the settings that actually yielded some improvements and omit negative results from [Table 1](#) below.

4. Linguistic Resources

4.1. ruWordNet as a dataset

RuWordNet is a lexical database in the WordNet format that was semi-automatically converted from RuThes-lite, a publicly available part of the linguistic ontology of the Russian language originally created for NLP applications [10]. According to the authors of RuThes-lite, it includes a subsection of the full database and comprises words that are particularly frequent in the news [9]. RuWordNet has inherited some of the properties of RuThes that are important for the task discussed here. It contains a considerable number of multi-word items, with many synsets having no single-word lemmas (unigrams) attached to them. The total number of all hyponym-hypernym pairs for nouns in the training data (provided by the organisers of the shared task) is 431,937; filtering out MWEs leaves us with 94,115 pairs. Note that polysemous words in ruWordNet are assigned to several synsets, and have different sets of hypernyms. However, our estimate, based on the training data, indicates that 76.9% of the listed hyponyms are monosemous.

4.2. Comparative Performance of Embeddings

We have tested nine available pre-trained embedding models, produced within different frameworks on raw and PoS-tagged Russian corpora of various size and make-up. Most models were obtained from the RusVectors repository [8]. Our observations indicated that (1) **fastText** models trained on the same corpora performed worse than **word2vec** (for example, MAP on the public test set (nouns) for **word2vec** trained on UD-tagged Russicum Maximum corpus and **fastText** vectors learnt on the lemmatised version of the same corpus compare as 0.2540 vs. 0.1406, even though **fastText** returns no OOV), and (2) lemmatised and tagged embeddings outperformed those trained on raw text (including extremely large vectors from Russian Distributional Thesaurus project [15]). The best performing vector model was a **word2vec** skipgram model trained on the Araneum Russicum Maximum corpus [1] (about 10 billion words). It significantly outperformed similar vectors (size 300, window 2) trained on concatenated Wikipedia and Russian National Corpus, the news corpus provided by the shared task organisers and on the Taiga corpus².

² https://tatianashavrina.github.io/taiga_site/

4.3. Extended MWE Support: Training our own Vectors

Co-occurrence statistics and Hearst patterns results were produced from a large corpus obtained by the concatenation of the four Russian corpora: Araneum Russicum Maximum, Russian Wikipedia, Russian National Corpus (both the main and newspaper parts), and the news corpus. All texts were lemmatised and PoS-tagged with UDPipe [18]. The overall size of the pre-processed corpus amounted to 900 million sentences, 8.7 billion tokens (after stop words removal). The same corpus was used to train customised embeddings with extended MWE support. To this end, we tokenised 39,000 noun phrases (NP) and 12,000 verb phrases (VP) from ruWordNet training data in this corpus, obtaining multi-word tokens like *'wild_ADJ::bird_NOUN'*. A standard CBOW embedding model was trained on the resulting corpus for three epochs, with a symmetric context window of size 2. Note that we aimed at covering as many ruWordNet lemmas as possible. For that, we set an unusually high vocabulary size of approximately 1 million top frequent word types (this essentially means each word with an absolute frequency of more than 60 received its vector). This measure rendered more than 85% of NP and 62% of VP from ruWordNet accessible and boosted the size of the effective training data for our classifier.

5. Experimental Setup

5.1. Projection Learning Approach

The baseline hypernym predictions for test words were obtained by finding (via exact normal equation solving) a linear projection matrix which minimises the sum of the squares of the differences between the target (hypernym) vectors and those returned by the multiplication of the projection matrix and the source (hyponym) vector; this largely follows the setup described in [7]. Experiments returned higher scores for normal equation solving without the regularisation term λ .

To learn projections with the loss function regularised by negative sampling using reversed hypernymy relations or synonyms, we adapted the code published by Ustalov et al. (2017) [19]. For the asymmetric regularisation we used the hyponym-hypernym word pairs from the provided training data; for synonyms as negative examples we used senses from the same synset as they appear in the training data.

5.2. Classification Approach

To obtain a model that predicts the hypernym synsets directly, we implemented a standard neural classifier. It uses the training pairs of hyponym word vectors and the associated hypernym synset ids. We discarded ids that appeared less than five times in the training data. The size of the training data, when using our own vectors with extended MWE support (see Section 4.3), is 74,235 train instances (hyponym words) and 4,068 classes (synsets) for nouns. The classifier has one hidden layer of size 386, a dropout of 0.1, a ReLu activation function and a softmax output

layer. It was trained with batch size 32, with the maximum number of epochs set to 25 (in practice, the training almost always stopped at around epoch 15, since the accuracy stopped increasing). At test time, the trained classifier was given a vector representation of a new hyponym and predicted the probability distribution over all possible classes (hypernym synset ids) for this vector. We return 10 synset ids with the highest probability values.

Note that the classifier was implemented as a shallow feed-forward neural network, but in fact, comparable results could be achieved with many other non-neural classifiers (logistic regression, etc). We chose a neural architecture because of the comparative ease of implementation.

6. Results

In Table 1, we report the mean average precision (MAP) scores for our methods described above on the private and public test sets (and the train-test split provided by the organisers) for nouns. Our best system (using tokenised MWE embeddings and the classifier approach) was eventually ranked 3rd (of 17 participants total) in the shared task leader board on the test set, with the MAP of 0.4976. The systems ranked 1st and 2nd showed MAP of 0.5522 and 0.5054 respectively. For verbs the performance of our approach is less competitive, with MAP of 0.2470 against the best result achieved in the competition of 0.4483.

Table 1: MAP scores for the methods we used and their modifications on noun test sets

| Approach | Araneum embeddings | | | embeddings with MWE | | |
|-------------------------------|--------------------|--------|----------|---------------------|--------|----------|
| | private | public | provided | private | public | provided |
| baseline projection | 0.2615 | 0.2470 | 0.2251 | 0.2846 | 0.2925 | 0.2693 |
| + Hearst patterns | 0.2653 | 0.2520 | 0.2298 | 0.2866 | 0.2942 | 0.2736 |
| neg-sampling synonymy [19] | 0.2604 | 0.2484 | 0.2384 | 0.2974 | 0.3153 | 0.2925 |
| + Hearst patterns | 0.2674 | 0.2506 | 0.2397 | 0.2996 | 0.3178 | 0.2959 |
| classifier | 0.4590 | 0.4219 | 0.3236 | 0.4976 | 0.4706 | 0.5105 |

The figures in the right-hand part of the table are consistently higher than in the left-hand part. The difference indicates the gains in performance attributable to the customised embeddings trained on the version of the corpus with the tokenised MWEs.

The results for several test sets indicate how stable the improvements are and link the performance to the specificity of the test set. In our scenario, this specificity is defined by the number of OOV items, as our solutions are contingent on the embeddings vocabulary size. With the standard vectors trained on Araneum Maximum, the number of OOV items was 3%, 4% and 22% for private, public and provided test sets respectively, which obviously affected the performance of projection-learning approaches. Our customised vectors have a higher coverage and close the gap between

the number of OOV among test words, returning only 0.26%, 0.66% and 5.66% for the same test sets.

Table 1 highlights that refining the order of predicted hypernyms within the range of top 15 and 50 hypernym-ids tuples with hypernyms extracted using Hearst patterns (for Araneum and customised vectors respectively) consistently improves the results, albeit marginally: most changes are in the third decimal digit. Enforcing projection learning with negative sampling based on synonyms secures additional small gains in most settings. We saw a significant precision growth in this task only when we fell back to a simpler and more straightforward classification approach, which required no fine-tuned thresholds or filtering.

7. Conclusion

This paper demonstrates the limited effectiveness of some traditional approaches to taxonomy enrichment based on Hearst patterns or projection learning. In our case, the most successful method to solve the given task of hypernym finding for Russian nouns was a comparatively simple classifier trained on static word embeddings of hyponyms as features and ruWordNet noun synset ids as classes, returning the 10 most probable synsets at test time for each query. The code and the word embedding model we used to solve the task are available at <https://github.com/kunilovskaya/hypohyper/>.

It is important that substantial gains in performance were obtained from using word embeddings trained on a large specifically tailored Russian corpus with MWEs from ru-WordNet merged into one token. This ensured better coverage of hyponym single-word and multi-word entities, and, as a result, significantly more training data for the classifier.

Other techniques which proved to be useful for us (but, eventually, not the best) were Hearst patterns and negative sampling with synonyms when learning the projection matrix.

References

1. *Benko, V.*: Aranea: Yet Another Family of (Comparable) Web Corpora Aranea: A Family of Comparable Gigaword Web Corpora View project Dictionary of Contemporary Slovak Language View project Aranea: Yet Another Family of (Comparable) Web Corpora. In: TSD 2014. pp. 257–264 (2014).
2. *Bojanowski, P. et al.*: Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 5, 135–146 (2017).
3. *Camacho-Collados, J. et al.*: SemEval-2018 Task 9: Hypernym Discovery. In: Proceedings of the 12th international workshop on semantic evaluation. pp. 712–724 (2018).
4. *Fu, R. et al.*: Learning Semantic Hierarchies via Word Embeddings. In: Proceedings of the 52nd annual meeting of the association for computational linguistics. pp. 1199–1209 Association for Computational Linguistics (2014).
5. *Hearst, M. A.*: Automatic acquisition of hyponyms from large text corpora. In: Proceedings of the 14th conference on computational linguistics-volume 2. pp. 539–545 Association for Computational Linguistics (1992).

6. *Jurgens, D., Pilehvar, M. T.*: SemEval-2016 Task 14: Semantic Taxonomy Enrichment. In: Proceedings of semeval-2016. pp. 1092–1102 (2016).
7. *Kutuzov, A. et al.*: Clustering comparable corpora of Russian and Ukrainian academic texts: Word embeddings and semantic fingerprints. In: Ninth workshop on building and using comparable corpora. pp. 3–10 (2016).
8. *Kutuzov, A., Kuzmenko, E.*: WebVectors: A toolkit for building web interfaces for vector semantic models. In: Ignatov, D. I. et al. (eds.) Analysis of images, social networks and texts: 5th international conference, aist 2016, yekaterinburg, russia, april 7–9, 2016, revised selected papers. pp. 155–161 Springer International Publishing, Cham (2017).
9. *Loukachevitch, N. et al.*: RuThes-lite, a publicly available version of Thesaurus of Russian language RuThes. In: Computational linguistics and intellectual technologies: Papers from the annual conference “dialogue”. pp. 340–349 (2014).
10. *Loukachevitch, N. et al.*: Creating Russian Wordnet by Conversion. In: Computational linguistics and intellectual technologies: Papers from the annual conference “Dialogue”. pp. 22–32 (2016).
11. *Mikolov, T. et al.*: Distributed representations of words and phrases and their compositionality. In: Burges, C. J. C. et al. (eds.) Advances in neural information processing systems 26. pp. 3111–3119 Curran Associates, Inc. (2013).
12. *Nikishina, I. et al.*: RUSSE’2020: Findings of the First Taxonomy Enrichment Task for the Russian Language. In: Computational linguistics and intellectual technologies: Papers from the annual conference “Dialogue”. (2020).
13. *Oakes, M. P.*: Using Hearst’s Rules for the Automatic Acquisition of Hyponyms for Mining a Pharmaceutical Corpus. In: RANLP text mining workshop. pp. 63–67 (2005).
14. *Panchenko, A. et al.*: TAXI at SemEval-2016 Task 13: a Taxonomy Induction Method based on Lexico-Syntactic Patterns, Substrings and Focused Crawling. In: Proceedings of semeval-2016, pp. 1320–1327 (2016).
15. *Panchenko, A. et al.*: Human and Machine Judgements for Russian Semantic Relatedness. Analysis of Images, Social Networks and Texts: 5th International Conference, AIST 2016, Yekaterinburg, Russia, April 7–9. 221–235 (2017).
16. *Sabirova, K., Lukanin, A.*: Automatic Extraction of Hypernyms and Hyponyms from Russian Texts. In: AIST (supplement). pp. 35–40 (2014).
17. *Sejr Schlichtkrull, M., Martínez Alonso, H.*: MSejrKu at SemEval-2016 Task 14: Taxonomy Enrichment by Evidence Ranking. In: Proceedings of semeval-2016. pp. 1337–1341 (2016).
18. *Straka, M., Straková, J.*: Tokenizing, POS Tagging, Lemmatizing and Parsing UD 2.0 with UDPipe. In: Proceedings of the conll 2017 shared task: Multilingual parsing from raw text to universal dependencies. pp. 88–99 (2017).
19. *Ustalov, D. et al.*: Negative sampling improves hypernymy extraction based on projection learning. In: Proceedings of the 15th conference of the European chapter of the association for computational linguistics: Volume 2, short papers. pp. 543–550 Association for Computational Linguistics, Valencia, Spain (2017).
20. *Yamane, J. et al.*: Distributional Hypernym Generation by Jointly Learning Clusters and Projections. In: Proceedings of coling 2016, the 26th international conference on computational linguistics: Technical papers. pp. 1871–1879 (2016).