

Meta-Embeddings in Taxonomy Enrichment Task

Tikhomirov M. M.

Lomonosov Moscow State University
Moscow, Russia
tikhomirov.mm@gmail.com

Loukachevitch N. V.

Lomonosov Moscow State University
Moscow, Russia
louk_nat@mail.ru

Abstract

In this paper we consider the taxonomy enrichment task based on a recently appeared dataset, called Diachronic wordnets, created on the basis of English and Russian wordnets. We study meta-embeddings approaches, which combine several source embeddings, to the hypernym prediction of novel words and show that meta-embedding approaches obtain the best results for this task if compared to other methods based on different principles. When combining with automatically extracted features from the Wiktionary online dictionary, the joint approach improves the results.

Key words: taxonomy enrichment, WordNet, meta-embeddings

DOI: 10.28995/2075-7182-2021-20-681-691

Мета-эмбединги в задаче пополнения таксономии

Тихомиров М. М.

МГУ имени М. В. Ломоносова
Москва, Россия
tikhomirov.mm@gmail.com

Лукашевич Н. В.

МГУ имени М. В. Ломоносова
Москва, Россия
louk_nat@mail.ru

Аннотация

В данной статье рассматривается задача обогащения таксономии на базе недавно появившегося набора данных, называемого Diachronic wordnets, созданного на основе английских и русских тезаурусов типа WordNet. Исследуется подход с использованием мета-эмбедингов, которые объединяют в себе разные векторные представления, для решения задачи предсказания гиперонимов. В статье показано, что подходы на основе мета-эмбедингов дают лучшие результаты на данной задаче по сравнению с другими методами, основанными на иных принципах. При комбинировании с автоматически извлеченными признаками из онлайн-словаря Викисловарь совместный результат становится еще лучше.

Ключевые слова: обогащение таксономии, ворднет, мета-эмбединги

1 Introduction

Various algorithms in natural language processing use vector representations of words, so the quality of the corresponding vectors is essential. There are many different word embeddings, which performance vary for various tasks. Different methods for constructing vectors capture the context in different ways, and can be trained on different datasets, resulting in a wide variety of vector models available. It has been shown that combining word embeddings can improve the accuracy of dependency parsing [2], classification in healthcare [20], named-entity recognition [31, 21], sentiment analysis [21].

Among lexical semantics settings, meta-embeddings were tested in word analogy and similarity tasks [33, 5, 4] but they were not applied to hypernym detection and taxonomy enrichment tasks, to the best of our knowledge.

Hyponym-hypernym relations ("is_a relations") constitute the backbone structure of many different ontological and lexical-semantic resources. Therefore numerous studies are devoted to the task of extracting hypernym relations from text collections. Hypernyms can be extracted from scratch, without any target resource or taxonomy. But also the hypernym extraction task can be set as a task of searching hypernyms for new words in an existing taxonomy, that is as a taxonomy enrichment task.

IN 2016, the taxonomy enrichment task was organized as a shared task at SemEval workshop (task 14) [15]. At this task, the participants should to attach words with definitions to correct hypernyms in WordNet [19]. However, in real applications definitions of novel words and their senses are most likely absent. In 2020, a new open evaluation on taxonomy enrichment of the Russian wordnet RuWordNet [6] RUSSE'2020 was organized [24]. The task was to find correct hypernyms from an older RuWordNet version for words described in a newer RuWordNet version. The work [29] describes new datasets called Diachronic wordnets¹ created on the basis of English and Russian wordnets. These datasets contain new words added to later versions of wordnets in comparison to earlier version together with their hypernyms in older versions. In such a way it is possible to use different versions of wordnets in their historical development to evaluate methods for the taxonomy enrichment task.

In this paper we show that meta-embedding approaches obtain the best results for the taxonomy enrichment task on the Diachronic-wordnets dataset if compared to other methods based on different principles. When combining with automatically extracted features from the Wiktionary online dictionary, the joint approach improves the results.

2 Related Work

2.1 Hypernym Detection Approaches

Traditional methods for hypernym detection include pattern-based methods, searching for specific hypernym patterns in sentences [14, 25, 26], methods based on similarity of word vector representations [17, 9], and also combined approaches integrating various context and similarity features of words [28, 3, 27].

In the RUSSE-2020 evaluation [24], in which the task was to predict RuWordNet hypernym synsets for new words, the participants used various word embeddings (static – fastText [10], word2vec [8], and contextualized - BERT [1]), the available RuWordNet taxonomy structure, hypernym and co-hyponym patterns, definitions of words from Wiktionary, and global search engines results [32, 7, 30, 24].

Recent methods to hypernym extraction exploit graph-based representations of taxonomy structure. Liu et al. [23] use node2vec embeddings of graph structures [13] for taxonomy induction. Aly et al. [12] use hyperbolic Poincare embeddings [22] for automatic generation of taxonomies. In [11], the authors study graph-based representation methods on the Diachronic-wordnets dataset.

2.2 Meta-Embeddings

The most simple vector combining approaches are concatenation or averaging of some source embeddings. The authors of [5] reported that even such simple combined embeddings could significantly improve the overall performance for several tasks. It has been shown [33] that using singular value decomposition (SVD) can also show good results with the ability to control the final dimension of vectors. Autoencoders [4], called Autoencoded Meta-Embeddings (AEME), became a further development of the idea of creating meta-embeddings. The authors of [4] proposed several different algorithms (CAEME, AAEME and etc) for combination various word vectors in one vector by encoding initial vectors in some meta-embedding space and then decoding backward.

At the first step in the CAEME approach, all word vectors are encoded into meta-vectors and then concatenated. Each vector is encoded into a vector of the same size, and the overall shape of the meta-vector is the sum of the original vector dimensions. Then, the decoding step uses a concatenated representation to predict the original vector representations. The AAEME approach is very similar to CAEME, except that each vector is mapped to a fixed-size vector and all encoded representations are averaged, but

¹<https://github.com/skoltech-nlp/diachronic-wordnets>

not concatenated. An obvious advantage of this approach is the ability to control the meta-embedding dimension.

For any AEME approach, different loss functions can be used at the decoding stage: MSE loss, KL-divergence loss, cosine distance loss and also their combinations. In [21] the authors investigated the performance of the autoencoders depending on the loss function. They found that there is no evident winner across tasks and that different loss functions should be chosen for different applications.

3 Datasets and Evaluation Measure

The Diacronic wordnets collection consists of two diachronic datasets: one for English, another one for Russian based respectively on Princeton WordNet [18] and RuWordNet taxonomies [6]. Each dataset contains a taxonomy and a set of novel words to be added to this resource. The statistics are provided in Table 1.

Table 1: Datasets statistics.

Dataset	Nouns	Verbs
<i>WordNet1.6 - WordNet3.0</i>	17 043	755
<i>WordNet1.7 - WordNet3.0</i>	6 161	362
<i>WordNet2.0 - WordNet3.0</i>	2 620	193
<i>RuWordNet1.0 - RuWordNet2.0</i>	14 660	2 154
<i>RUSSE'2020</i>	2 288	525

To compile the English dataset, two versions of WordNet were chosen and then words, which appear only in a newer version, were selected. For each novel word, its hypernyms from the newer WordNet version were extracted and considered as gold standard hypernyms. Novel words were added to the dataset if only their hypernyms appear in both versions. Only nouns and verbs were considered. Several datasets by skipping one or more WordNet versions were created (Table 1).

As gold standard hypernyms, not only the immediate hypernyms of each lemma were considered but also the second-order hypernyms: hypernyms of the hypernyms. The aim was to make the evaluation less restricted but to keep it quite precise.

In order to create an analogous version to English dataset for Russian, a Russian WordNet counterpart, the RuWordNet taxonomy [6] was used. The current version of RuWordNet and also the extended version of RuWordNet, which has not been published yet, were taken to compile the dataset (cf. Table 1).

Another variant of the Russian dataset includes the dataset created for RUSSE'2020 Dialog evaluation [24]. The RUSSE'2020 can be viewed as a restricted subset of the Russian dataset, because of exclusion from it several word categories (short words, diminutive forms, geographic and personal names, and others).

The task of thesaurus enrichment is treated as a ranking task where the correct answers should be in the top of a candidate list. For evaluation, a traditional measure for ranking tasks Mean Average Precision measure is used.

$$MAP = \frac{1}{N} \sum_{i=1}^N AP_i; \quad (1)$$

$$AP_i = \frac{1}{M} \sum_{i=1}^n prec_i \times I[y_i = 1],$$

where N and M are the number of predicted and ground truth values, respectively, $prec_i$ is the fraction of ground truth values in the predictions from 1 to i , y_i is the label of the i -th answer in the ranked list of predictions, and I is the indicator function.

To account for second-order hypernyms, the modified MAP measure is used. It transforms a list of gold standard hypernyms into a list of connected components. Each of these components includes hypernyms (both direct and second-order) which form a connected component in a taxonomy graph. To obtain the highest score, the model should guess a hypernym from each connectivity component [24].

4 Creating Meta-Embeddings

In our work we compare simple meta-embeddings such as concatenation of source embeddings and SVD over the concatenation and two variants of autoencoders generating meta-embeddings: Concatenated Autoencoded Meta-Embeddings (CAEME) and Averaged Autoencoded Meta-Embeddings (AAEME), which have shown good results in previous works [4].

Suppose we have two source embeddings $s_1(w)$ and $s_2(w)$, their encoders $E_1(w)$ and $E_2(w)$ and their decoders $D_1(w)$ and $D_2(w)$. Meta-embedding $m(w)$ in CAEME is constructed as the L_2 -normalised concatenation of two encoded source embeddings $E_1(s_1(w))$ and $E_2(s_2(w))$:

$$m(w) = \frac{E_1(s_1(w)) \oplus E_2(s_2(w))}{\|E_1(s_1(w)) \oplus E_2(s_2(w))\|_2} \quad (2)$$

where \oplus is the concatenation operation.

In CAEME, the dimensionality of the meta-embedding space is the sum of the dimensions of the source embeddings. The AAEME encoder can be seen as a special case of the CAEME encoder, where the meta-embedding is computed by averaging the two encoded sources in (2) instead of their concatenation. Averaging gives the possibility to avoid increasing the dimensionality of the meta-embedding.

The AAEME encoder computes the meta-embedding of a word w from its two source embeddings $s_1(w)$ and $s_2(w)$ as the L_2 -normalised sum of two encoded versions of the source embeddings $E_1(s_1(w))$ and $E_2(s_2(w))$.

$$m(w) = \frac{E_1(s_1(w)) + E_2(s_2(w))}{\|E_1(s_1(w)) + E_2(s_2(w))\|_2} \quad (3)$$

The CAEME and AAEME decoders reconstruct the source embeddings from the same meta-embedding $m(w)$, thereby implicitly using both common and complementary information in the source embeddings. The constructed source embeddings look as follows:

$$\begin{aligned} \hat{s}_1(w) &= D_1(m(w)) \\ \hat{s}_2(w) &= D_2(m(w)) \end{aligned} \quad (4)$$

The overall objective of autoencoder training is given below. Function f can be any distance or similarity measure as MSE, KL-divergence, or cosine distance. The coefficients λ_1 and λ_2 can be used to give different emphasis to the reconstruction of the two sources.

$$Loss_w(E_1, E_2, D_1, D_2) = \sum_w (\lambda_1 f(s_1(w), \hat{s}_1(w)) + \lambda_2 f(s_2(w), \hat{s}_2(w))) \quad (5)$$

Jointly learning of E_1, E_2, D_1, D_2 minimises the total reconstruction error given by Equation 5.

To obtain meta-embedding representations after training, only the encoders are applied, which convert the input source embeddings into a meta representation. Further, these meta-embedding vectors are used as vector representations of words.

5 Meta-Embeddings in Taxonomy Enrichment Task

In our approach, we use embeddings (source embeddings or meta-embeddings) to generate a list of most similar taxonomy entries (words or phrases from the taxonomy) to the target word according to cosine similarity. For each target word, the top 20 taxonomy entries are considered. The number of elements for consideration was chosen experimentally. For each entry in the similarity list, all correspondings synsets, their direct and second-order hypernyms are extracted from the taxonomy. They are considered as candidate synsets to be hypernyms of the target word.

For candidate hypernyms synsets, several features are calculated. Logistic regression is used to predict the probability of a candidate to be a hypernym of the target word. The calculated features are as follows:

- maximal, minimal and average similarity between the target word and synonyms in a candidate synset;

- similarity values between the the target word and synonyms in hyponym synsets of the candidate synset. At first, maximal, minimal and average similarity values are calculated for the target and synonyms in each hyponym synset. Then maximal, minimal and average similarity values are calculated over all hyponym synsets for a given candidate synset;
- positional feature (0, 1, 2): if a candidate is one of the synset of a taxonomy entry or it is its direct or second-order hypernym.
- the number of occurrences of the synset in the candidate list.

In total, 17 features were calculated.

Training data were generated randomly and automatically from the published RuWordNet version and WordNet-1.6 (thus, the data do not contain test data).

Table 2: Training Datasets sizes

Language	Part of speech	
	Nouns	Verbs
English	2931	1532
Russian	2990	814

Table 3: MAP scores for the taxonomy enrichment methods for the English datasets.

method	nouns			verbs			vector dim
	1.6-3.0	1.7-3.0	2.0-3.0	1.6-3.0	1.7-3.0	2.0-3.0	
fastText	0.300	0.346	0.396	0.290	0.224	0.280	300
word2vec	0.226	0.242	0.265	0.091	0.114	0.150	300
GloVe	0.261	0.290	0.326	0.182	0.145	0.175	300
concat	0.308	0.344	0.387	0.273	0.206	0.247	900
SVD	0.308	0.358	0.406	0.286	0.222	0.271	600
CAEME	0.309	0.347	0.395	0.252	0.189	0.260	900
CAEME triplet loss	0.322	0.367	0.416	0.287	0.218	0.270	900
AAEME	0.318	0.354	0.401	0.283	0.218	0.254	600
AAEME triplet loss	0.333	0.373	0.416	0.289	0.227	0.274	600
Previous results based on embeddings:							
fastText [29]	–	–	0.339	–	–	0.213	–
Poincaré embeddings[12, 11]	0.059	0.066	0.101	0.126	0.066	0.109	–
node2vec [13, 11]	0.194	0.219	0.155	0.151	0.109	0.147	–
GCN autoencoder [16, 11]	0.157	0.175	0.168	0.109	0.094	0.117	–

The quality of the approach was evaluated using different source vector representations: fastText², word2vec³, GloVe⁴. Also different meta-embedding approaches were investigated: concatenation, SVD over concatenation, CAEME, AAEME. The standard loss function for AEME approaches we used was cosine distance loss. We have tried variations and combinations between MSE loss, KL divergence loss and cosine distance loss, and last one works best in our case.

A specific feature of the English datasets is the presence of a significant number of multi-word expressions. In order to obtain vectors for such cases, the following procedures were carried out:

- For FastText, embeddings if not in the vocabulary, were obtained in a natural way, by calculating vectors by the model itself,
- For Word2Vec and GloVe, embeddings were calculated by averaging the vectors of maximum prefixes for the constituent words of a multi-word expressions. There is a limitation on the minimum length of a prefix word, which is 4 characters,

²Common Crawl English and Russian versions from <https://fastText.cc/docs/en/crawl-vectors.html>

³Araneum for Russian and Gigaword for English from <http://vectors.nlp.eu/repository/>

⁴Common Crawl 840b tokens from <https://nlp.stanford.edu/projects/GloVe/>

- For meta-embeddings approaches, if there is no vector for a word in any model, the corresponding source vector was initialized with zeros.

Another direction of experiments relates to the additional restrictions to the generated meta-embeddings in the AEME algorithms such as the triplet loss. We suppose that the word should be closer to its semantically related words according to the taxonomy than to a randomly chosen word. The algorithm of calculating the triplet loss is as follows:

1. For each word that is present in the taxonomy, a list of semantically related words is compiled from synonyms, hyponyms and hypernyms,
2. In each epoch, we randomly select K positive words from this related set and K negative words from the vocabulary,
3. If the word is not presented in the taxonomy, then the noisy variations of the original vector are positive vectors,
4. Next, triplet margin loss is calculated: the triplet loss is combined with the original loss as $\alpha * loss + (1 - \alpha) * triplet_loss$, we used $\alpha = 0.5$,

The results of the experiments are given in Table 3 and Table 4. It can be seen that the hypernym prediction for verbs is much worse than for nouns in all datasets. SVD applied to the concatenation of initial embeddings always improves the results compared to the concatenation. It obtains better results if compared to source embeddings in most datasets, except English verbs. SVD achieves the best quality of hypernym prediction among all considered meta-embeddings on the Russian verb datasets.

Table 4: MAP scores for the taxonomy enrichment methods for the Russian datasets.

method	nouns		verbs		vector dim
	non-restricted	restricted	non-restricted	restricted	
fastText	0.416	0.537	0.318	0.418	300
word2vec	0.276	0.526	0.231	0.272	600
concat	0.401	0.563	0.337	0.423	900
SVD	0.435	0.579	0.382	0.442	600
CAEME	0.468	0.579	0.352	0.433	900
CAEME triplet loss	0.470	0.580	0.350	0.420	900
AAEME	0.466	0.577	0.350	0.432	600
AAEME triplet loss	0.474	0.581	0.375	0.439	600
Previous results based on embeddings:					
RUSSE Top-1 for verbs: [7]	0.288	0.418	0.340	0.448	–
Poincaré embeddings [12, 11]	0.143	0.252	0.105	0.140	–
node2vec [13, 11]	0.266	0.366	0.168	0.252	–
GCN autoencoder [16, 11]	0.183	0.261	0.095	0.141	–

The AAEME autoencoder with the cosine loss is comparable to the CAEME autoencoder in the Russian data, and better on the English datasets. Among meta-embeddings based on autoencoders, the best results of hypernym prediction are obtained with the AAEME autoencoder with triplet loss. This approach improves the results of hypernym prediction in almost all cases compared to initial fastText embeddings, except English verbs. Triplet loss autoencoders in most cases achieve better results than corresponding cosine loss autoencoders.

If compared to other approaches based only on various vector representations, our applied methods obtained the best results on all datasets, except Russian restricted verbs where the results are slightly worse.

6 Improving Meta-Embeddings Results Using Wiktionary

Some previous approaches [32, 24] were tested in hypernym prediction combining embeddings with automatic Wiktionary analysis.

Each Wiktionary page usually comprises a definition and lists of hypernyms, hyponyms and synonyms, which could be useful for our task. Similar to other approaches, we implement the following Wiktionary

features:

- the candidate is present in the Wiktionary hypernyms list for the input word (binary feature),
- the candidate is present in the Wiktionary synonyms list (binary feature),
- the candidate is present in the Wiktionary definition (binary feature),

We do not use the definitions directly, as their texts are too noisy. They often include example usages of words, which cannot be separated from the definitions and can distort their vector representations. This processing generate additional above-mentioned features, which are added to the described logistic regression classifier.

Tables 5 and 6 shows the results of combined approaches. Adding Wiktionary features improved the achieved results on all datasets. If compared to previous approaches used Wiktionary, we obtained the best results on the Diachronic wordnet datasets.

Table 5: MAP scores for the taxonomy enrichment methods for the English datasets with wiktionary features.

method	nouns			verbs		
	1.6-3.0	1.7-3.0	2.0-3.0	1.6-3.0	1.7-3.0	2.0-3.0
fastText	0.319	0.373	0.424	0.296	0.231	0.288
word2vec	0.236	0.259	0.288	0.090	0.12	0.149
GloVe	0.277	0.315	0.357	0.189	0.153	0.190
concat	0.326	0.370	0.425	0.275	0.213	0.259
SVD	0.324	0.380	0.437	0.289	0.228	0.276
CAEME	0.327	0.373	0.430	0.262	0.205	0.264
CAEME triplet loss	0.339	0.39	0.44	0.292	0.225	0.277
AAEME	0.334	0.379	0.432	0.286	0.225	0.267
AAEME triplet loss	0.345	0.394	0.445	0.289	0.239	0.272
Previous results with wiki:						
fastText[29]	0.337	0.380	0.344	0.267	0.200	0.237
fastText + node2vec [11]	0.313	0.380	0.340	0.259	0.195	0.200
fastText+node2vec + Poincaré [11]	0.311	0.350	0.300	0.251	0.177	0.248

Table 6: MAP scores for the taxonomy enrichment methods for the Russian datasets with wiktionary features (wiki).

method	nouns		verbs	
	non-restricted	restricted	non-restricted	restricted
fastText	0.436	0.579	0.366	0.445
word2vec	0.292	0.574	0.258	0.295
concat	0.421	0.598	0.388	0.474
SVD	0.452	0.612	0.423	0.458
CAEME	0.485	0.614	0.400	0.472
CAEME triplet loss	0.486	0.607	0.402	0.463
AAEME	0.484	0.611	0.403	0.486
AAEME triplet loss	0.490	0.611	0.427	0.471
Previous results with wiki:				
RUSSE Top-1 for nouns [29, 24]	0.393	0.552	0.293	0.436
fastText+wiki [29]	0.413	0.551	0.297	0.389
fastText + node2vec + Poincaré+wiki [11]	0.414	0.560	0.306	0.391

7 Analysis of Results

It can be seen from the achieved results that having a large taxonomy, automatic methods are able to predict correct taxonomy hypernyms for novel words within three first positions of the ranked list of candidates on average. For the Russian dataset, the correct predictions could be found among the top-2 candidates on average. For the majority of novel words, the correct hypernyms appear within the top-10 candidates.

To analyze the usefulness of the obtained hypernym predictions, we calculated the proportions of words having at least one correct hypernym withing N first positions of the candidate list. Table 7 shows the obtained proportions of words with the first correct answer at Top-N positions.

It can be seen that current results do not allow using for fully automatic taxonomy enrichment because the predictions of correct answers on the first position are still quite low. But the predicted results can significantly facilitate the work of lexicographers or knowledge engineers.

We analysed hypernym predictions for words not included in the Top-10 of correct answers and found the following cases:

- Predicted hypernyms correspond to senses missed in the taxonomy. For example, word *vechernitsa* is described in RuWordNet only in the sense of 'student of evening education', but also this word can mean 'bat' (noctule bat) or 'flowering plant'. Predicted hypernyms include synset corresponding to the flower synset at the 3d position of the candidate list;
- in many cases predictions are very semantically close but not correct. For example, for word 'gugl' (Google) the correct answers are synsets 'search engine' and 'global search engine'. The predicted hypernyms are 'computer program', 'internet-technology', 'internet-site', 'IT-technology', etc. For word "datacentr" (data-center), which is a specialized building, the prediction at the first position is the 'computer' synset.
- there are also numerous examples when too general hypernyms are predicted;
- in some cases predicted hypernyms are very far from reasonable answers and are difficult for explanation.

Table 7: First correct answer at top N positions. The results are give for nouns based on AAEME triplet-loss embeddings with Wiktionary.

	English %			Russian %	
	1.6-3.0	1.7-3.0	2.0-3.0	non-restricted	restricted
Top-1	28	32	38	42	52
Top-3	41	45	51	57	75
Top-5	47	52	56	64	81
Top-10	54	59	63	70	87

8 Conclusion

In this paper we considered methods for combining different word embeddings in a single meta-embedding. We studied the meta-embedding approach in the taxonomy enrichment task based on several versions of English and Russian wordnets. The meta-embedding methods included concatenation of initial embeddings, SVD over the concatenation, two variants of autoencoders aimed to learn better word embeddings from initial vectors.

We showed that the use of meta-embeddings improves the performance of the system for almost all datasets, except English verbs. SVD always improves the results compared to concatenation. Autoencoder-based meta-embeddings achieve the best results in most cases. It can also be seen that adding the triplet loss improves the results.

We also experimented with a method of joint using meta-embeddings and information about word described in the Wiktionary electronic dictionary, which improved the results.

Acknowledgements

The participation of M. Tikhomirov in the reported study was funded by RFBR, project number 19-37-90119. The work of Natalia Loukachevitch in the current study (preparation of data for the experiments) is supported by the Russian Science Foundation (project 20-11-20166).

References

- [1] BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding / Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova // Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). — Minneapolis, Minnesota : Association for Computational Linguistics, 2019. — Jun. — P. 4171–4186. — Access mode: <https://www.aclweb.org/anthology/N19-1423>.
- [2] Bansal Mohit, Gimpel Kevin, Livescu Karen. Tailoring continuous word representations for dependency parsing // Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). — 2014. — P. 809–815.
- [3] Bernier-Colborne Gabriel, Barriere Caroline. Crim at semeval-2018 task 9: A hybrid approach to hypernym discovery // Proceedings of the 12th international workshop on semantic evaluation. — 2018. — P. 725–731.
- [4] Bollegala Danushka, Bao Cong. Learning word meta-embeddings by autoencoding // Proceedings of the 27th international conference on computational linguistics. — 2018. — P. 1650–1661.
- [5] Coates Joshua, Bollegala Danushka. Frustratingly Easy Meta-Embedding–Computing Meta-Embeddings by Averaging Source Word Embeddings // arXiv preprint arXiv:1804.05262. — 2018.
- [6] Creating Russian wordnet by conversion / Natalia V Loukachevitch, German Lashevich, Anastasia A Gerasimova et al. // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”. — 2016. — P. 405–415.
- [7] Dale David. A simple solution for the Taxonomy enrichment task: Discovering hypernyms using nearest neighbor search // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”. — 2020.
- [8] Distributed Representations of Words and Phrases and their Compositionality / Tomas Mikolov, Ilya Sutskever, Kai Chen et al. // Advances in Neural Information Processing Systems 26 / Ed. by C. J. C. Burges, L. Bottou, M. Welling et al. — Curran Associates, Inc., 2013. — P. 3111–3119.
- [9] Do supervised distributional methods really learn lexical inference relations? / Omer Levy, Stefan Remus, Chris Biemann, Ido Dagan // Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. — 2015. — P. 970–976.
- [10] Enriching Word Vectors with Subword Information / Piotr Bojanowski, Edouard Grave, Armand Joulin, Tomas Mikolov // Transactions of the Association for Computational Linguistics. — 2017. — Vol. 5. — P. 135–146.
- [11] Evaluation of Taxonomy Enrichment on Diachronic WordNet Versions . / Irina Nikishina, Alexander Panchenko, Varvara Logacheva, Natalia Loukachevitch // Proceedings of the 11th Global WordNet conference GWC-2021. — 2021.
- [12] Every Child Should Have Parents: A Taxonomy Refinement Algorithm Based on Hyperbolic Term Embeddings / Rami Aly, Shantanu Acharya, Alexander Ossa et al. // Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. — Florence, Italy : Association for Computational Linguistics, 2019. — Jul. — P. 4811–4817. — Access mode: <https://www.aclweb.org/anthology/P19-1474>.
- [13] Grover Aditya, Leskovec Jure. node2vec: Scalable feature learning for networks // Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. — 2016. — P. 855–864.

- [14] Hearst Marti A. Automatic acquisition of hyponyms from large text corpora // *Coling 1992 volume 2: The 15th international conference on computational linguistics*. — 1992.
- [15] Jurgens David, Pilehvar Mohammad Taher. SemEval-2016 Task 14: Semantic Taxonomy Enrichment // *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*. — San Diego, California : Association for Computational Linguistics, 2016. — Jun. — P. 1092–1102. — Access mode: <https://www.aclweb.org/anthology/S16-1169>.
- [16] Kipf Thomas N, Welling Max. Semi-supervised classification with graph convolutional networks // *arXiv preprint arXiv:1609.02907*. — 2016.
- [17] Learning semantic hierarchies via word embeddings / Ruiji Fu, Jiang Guo, Bing Qin et al. // *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. — 2014. — P. 1199–1209.
- [18] Miller George A. WordNet: a lexical database for English // *Communications of the ACM*. — 1995. — Vol. 38, no. 11. — P. 39–41.
- [19] Miller George A. WordNet: An electronic lexical database. — MIT press, 1998.
- [20] Mixed Pooling Multi-View Attention Autoencoder for Representation Learning in Healthcare / Shaika Chowdhury, Chenwei Zhang, Philip S Yu, Yuan Luo // *arXiv preprint arXiv:1910.06456*. — 2019.
- [21] Neill James O', Bollegala Danushka. Meta-embedding as auxiliary task regularization // *arXiv preprint arXiv:1809.05886*. — 2018.
- [22] Nickel Maximilian, Kiela Douwe. Poincaré embeddings for learning hierarchical representations // *arXiv preprint arXiv:1705.08039*. — 2017.
- [23] On interpretation of network embedding via taxonomy induction / Ninghao Liu, Xiao Huang, Jun-dong Li, Xia Hu // *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. — 2018. — P. 1812–1820.
- [24] RUSSE'2020: Findings of the First Taxonomy Enrichment Task for the Russian Language / Irina Nikishina, Varvara Logacheva, Alexander Panchenko, Natalia Loukachevitch // *Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”*. — 2020.
- [25] Roller Stephen, Kiela Douwe, Nickel Maximilian. Hearst Patterns Revisited: Automatic Hypernym Detection from Large Text Corpora // *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. — 2018. — P. 358–363.
- [26] Sabirova Kristina, Lukanin Artem. Automatic Extraction of Hypernyms and Hyponyms from Russian Texts. // *AIST (Supplement)*. — 2014. — P. 35–40.
- [27] Shwartz Vered, Dagan Ido. Path-based vs. Distributional Information in Recognizing Lexical Semantic Relations // *COLING 2016*. — 2016. — P. 24.
- [28] Snow Rion, Jurafsky Dan, Ng Andrew Y. Semantic taxonomy induction from heterogenous evidence // *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. — 2006. — P. 801–808.
- [29] Studying Taxonomy Enrichment on Diachronic WordNet Versions / Irina Nikishina, Alexander Panchenko, Varvara Logacheva, Natalia Loukachevitch // *Proceedings of the 28th International Conference on Computational Linguistics*. — Barcelona, Spain : Association for Computational Linguistics, 2020. — December.
- [30] Tikhomirov Mikhail, LOukachevitch Natalia, Ekaterina Parkhomenko. Combined approach to hypernym detection for thesaurus enrichment // *Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”*. — 2020.
- [31] Winata Genta Indra, Lin Zhaojiang, Fung Pascale. Learning multilingual meta-embeddings for code-switching named entity recognition // *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*. — 2019. — P. 181–186.

- [32] Word2vec not dead: predicting hypernyms of co-hyponyms is better than reading definitions / Nikolay Arefyev, Maksim Fedoseev, Andrey Kabanov, Vadim Zizov // Computational Linguistics and Intellectual Technologies: papers from the Annual conference “Dialogue”. — 2020.
- [33] Yin Wenpeng, Schütze Hinrich. Learning word meta-embeddings // Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). — 2016. — P. 1351–1360.