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

Moscow, June 17–20, 2020

A SIMPLE SOLUTION FOR THE TAXONOMY ENRICHMENT TASK: DISCOVERING HYPERNYMS USING NEAREST NEIGHBOR SEARCH

Dale D. S. (dale.david@yandex.ru)
Yandex, Moscow, Russia

In this paper, we present the system we used in the Taxonomy Enrichment for the Russian Language evaluation campaign. The goal of this challenge is to predict hypernyms for the words not included in the taxonomy. Our approach was to generate and score candidate hypernyms by word embedding similarity of the input words and concepts already in the taxonomy. Despite being very simple, our system was ranked first on the verbs track.

Keywords: wordnet, hypernym, taxonomy enrichment, word2vec, nearest neighbor

DOI: 10.28995/2075-7182-2020-19-177-186

ПРОСТОЕ РЕШЕНИЕ ЗАДАЧИ ПО ОБОГАЩЕНИЮ ТАКСОНОМИИ: ВЫЯВЛЕНИЕ ГИПЕРОНИМОВ С ПОМОЩЬЮ ПОИСКА БЛИЖАЙШИХ СОСЕДЕЙ

Дале Д. С. (dale.david@yandex.ru)
Яндекс, Москва, Россия

В этой статье мы представляем систему, использованную нами в соревновании по обогащению таксономии для русского языка. Задача соревнования — предсказать гиперонимы слов, не включённых в таксономию. Для этого мы генерируем и ранжируем гиперонимы-кандидаты по сходству словных эмбеддингов входных слов с эмбеддингами понятий, уже включённых в таксономию. Несмотря на свою простоту, наша система достигла наилучшей точности на подзадаче поиска гиперонимов для глаголов.

Ключевые слова: wordnet, гиперонимы, обогащение таксономии, word2vec, метод ближайших соседей

1. Introduction

Hypernymy is the name for “is a” relation between words or phrases: e.g. hypernym of “whale” is “sea mammal”, and hypernyms of “sea mammal” are “mammal” and “sea creature”. Thesauri labeled with hypernymy relation can be used to solve tasks such as resolution of lexical ambiguity [11], query expansion in information retrieval [6], [11], processing questions and answers in question answering systems [6], [11], sentiment analysis and semantic similarity measurement [9], etc. One of such databases, WordNet [14] for the English language, has been in use for more than 20 years and remains a valuable source for various applications [9]. However, manually producing hypernyms for new words is time-consuming and expensive [9]. Therefore automatic discovery of hypernyms is an important problem [3], [4], [6], [9].

The evaluation campaign “Taxonomy Enrichment for the Russian Language” organized by the international conference “Dialogue 2020”1 [15] in which we take part is aimed exactly at this problem. Its goal is to provide 10 ranked candidate hypernyms for each new word in the test set. Hypernyms should be chosen from the existing RuWordNet taxonomy [12]. The challenge consists of two separate tracks for nouns and for verbs.

We approach the problem of hypernymy discovery by exploiting the existing structure of RuWordNet. This thesaurus contains 85K (33K) terms grouped into 30K (7K) synsets for nouns (verbs), and we expect2 that most new words have siblings (i.e. terms with the same hypernyms) in RuWordNet. The siblings should be semantically close to each other, so we expect that their word embeddings are also similar. Therefore, we use a weighted K-nearest-neighbor algorithm over word embeddings to retrieve potential siblings and rank their hypernyms as potential hypernyms of the query term.

This simple algorithm turned out to be unexpectedly effective, and we managed to achieve the best score for verbs track with it. In this paper, we describe it in more detail and analyze what makes our approach successful.

[1] http://www.dialog-21.ru/evaluation/

[2] It turns out to be true; see the subsection “Siblings”.

2. Related Work

Two important approaches to hypernymy discovery are pattern-based and distributional [3]. The pattern-based approach pioneered by Hearst [8] predicts hypernymy between words if they often co-occur in patterns like "A, such as B". The distributional approaches make use of distributional representations of terms, such as word embeddings [7]. Another important line of work utilizes definitions of terms, instead of unstructured corpora, to propose hypernyms for the terms [9].

Biemann et al. [4] give a good overview of existing approaches for enriching lexical semantic resources with distributional data. They also provide their own system for building taxonomies based on graphs of semantically related words induced from corpora.

Despite the importance of the hypernymy discovery problem, the challenge "Taxonomy Enrichment for the Russian Language" [15] seems to be the first campaign for Russian or any other Slavic language that evaluates discovery of hypernyms for new terms. However, there were similar competitions for English and other European languages, most notably SemEval-2016 Task 14 [9] (enriching a taxonomy using the definitions of the new words) and SemEval-2018 Task 9 [6] (extracting hypernyms from unlabeled corpora).

Best solutions of SemEval-2018 Task 9 include CRIM [2] (pattern-based discovery and scoring query-hypernyms pairs with a neural net), 300-sparsans [1] (sparse features and formal concept analysis). In SemEval-2016 Task 14, the winning system was MSerjKu [16] (classification of query-hypernym pairs using SVM with distributional and linguistic features).

3. Task description

3.1. Goal and metrics

The task is formulated as follows: for each term (query) in the test set, one should provide a list of 10 candidate hypernyms. They are evaluated against the ground truth: human-labeled hypernyms and hypernyms of these hypernyms. All these first- and second-order hypernyms are divided into connected components, and ranking scores are evaluated relatively to these components. The scores include mean average precision[3] (MAP) at the true number of hypernym components, mean reciprocal rank[4] (MRR) at 10, and F1 score (at the top 1 prediction); the official metric is MAP. The formulas for calculation MRR and MAP were customized to treat the whole connected component of hypernyms as a single hypernym. They are available in the official repository of the competition[5].

The task includes two separate tracks for nouns and verbs.

[3] https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval)#Mean_average_precision

[4] https://en.wikipedia.org/wiki/Mean_reciprocal_rank

[5] Formulas for both MAP and MRR are in the file **evaluate.py** in https://github.com/dialogue-evaluation/taxonomy-enrichment.

3.2. Provided datasets

The main piece of the provided data is the RuWordNet taxonomy, and train/test sets based on it. Some more additional datasets were proposed, but we did not use them.

RuWordNet is a collection of synsets (sets of terms with the same meaning) and relations (such as hypernymy/hyponymy) between them. Each synset consists of the terms (which may be single- or multi-word expressions), the title, and (optionally) the definition, and has a unique identifier.

The train set includes 25K nouns and 7K verbs with their first- and second-order hypernyms grouped into connected components. The public test set includes 762 nouns and 175 verbs, and the private test set—1525 nouns and 350 verbs without any labels.

3.3. Data split

All our experiments were evaluated on the dev set (randomly selected 2% of the training set), and on the public test set. There are two reasons, why these scores may be mismatched. The first reason is that nearly 74% nouns and 80% verbs from the training set have at least one sense in the existing RuWordNet taxonomy, whereas the test set has no intersection with the taxonomy. And some of these intersections are inconsistent: for example, **ОТКРОВЕННОСТЬ** (openness) has a sense **ЧИСТОСЕРДЕЧНЫЙ** (sincere) in the taxonomy but does not have corresponding hypernyms in the training set. To restrict the influence of this discrepancy, we exclude from the taxonomy all the terms in the dev set and their direct synonyms when we evaluate on the dev set. Second, 70% of nouns and 60% of verbs in the training set are in fact multi-word phrases such as **МУСКАТНОЕ ВИНО** (muscat wine), whereas the test sets consist only of sole words.

3.4. Siblings

Our chief hypothesis that most new words have siblings in RuWordNet proved itself true. In the training dataset, 90% of nouns and 99% of verbs have siblings. Moreover, 99.98% of training nouns and 100% of training verbs have either siblings or “cousins” (terms with common second-order hypernyms). Words without siblings include some neologisms (e.g. **ПОЛИТТЕХНОЛОГ** is the only hyponym of **ИМИДЖМЕЙКЕР**) and rare toponyms (e.g. **ЮГРА** is the only hyponym of **АВТОНОМНЫЙ ОКРУГ РФ**).

4. System description

The proposed method of predicting hypernyms is based on cosine similarity between fixed (context-independent) term embeddings. For each new term, we find its *k* nearest neighbors among the terms in the taxonomy and use their first- and second-order hypernyms as candidate hypernyms.

4.1. Index construction

We construct the pool of potential neighbors by taking for each RuWordNet synset its title, all its senses, and a concatenation of its title and its senses. For each of these texts, we calculate its text embedding as a weighted mean embedding of all words in it. The weights of the words in our implementation depend only on POS tags, but in a more complex setting, they could be tied e.g. to the syntactic role of the word. We L2-normalize word embeddings before aggregation in order to make representation of words more comparable to each other. We also L2-normalize sentence embeddings after aggregation in order to make Euclidean distance between them equivalent to cosine distance and simplify neighbor search.

To extract word embeddings, we use a word2vec [13] model pretrained on the Taiga corpus [17] and published on RusVectores [10].[6] Before lookup, we lemmatize each word and append the POS label to it. If the word is missing in the vocabulary of this model, we find all words in the vocabulary with the longest prefix matching this word and compute its embedding as the mean of their embeddings. For example, the embedding of the word **перуанка _ NOUN** (a female Peruvian) is computed as the mean of embeddings of **перуанец _ NOUN** (a male Peruvian) and **перуанский _ ADJ** (Peruvian).

As an alternative way to extract word embeddings, we use a fastText [5] model, which was pretrained on Taiga and published on RusVectores as well. It differs from the model above in two ways: it does not include POS tags, and it constructs embeddings for unseen words by averaging the embeddings of their character n-grams.

4.2. Ranking candidates

For each query term, we find its k nearest neighbors in the index (using the embeddings described above), and use all the first- and second-order hypernyms of the neighbors' synsets as answer candidates. We score occurrences of hypernyms with each particular neighbor separately and add together such scores for each hypernym candidate. The resulting prediction is the 10 candidate hypernyms with the highest total scores.

The score for each hypernym associated with a particular neighbor is calculated as

$$score = exp\left(-d^{\alpha}\right) \times s^{\beta} \times \begin{cases} 1, & \text{for first-order hypernyms of the neighbor} \\ \gamma, & \text{for second-order hypernyms} \end{cases}$$

where s is cosine similarity between the query and the neighbor, and $d = \sqrt{2\,(1-s)}$ is the distance between them. The constant γ reflects the preference between first- and second-order hypernyms. This formula was constructed manually and performed no worse than our attempt to train linear scoring formulas on the training datasets. In fact, the functions $exp(-d^{\alpha})$ and s^{β} have similar shapes, and only one of them would suffice, but we kept both to make the formula more flexible (and as a legacy of our experiments).

In general, with this formula we try to combine the evidence from the few close neighbors with the evidence from numerous distant neighbors. The parameters α and

[6] The model can be downloaded from https://rusvectores.org/ru/models/.

β are tuned in order to balance these signals. High values of α and β decrease the impact of the neighbors which are far from the query, allowing to use higher values of k, i.e. evidence from more neighbors.

5. Experiments and results

After some preliminary experiments, we chose and submitted the solution with $k = 100$, $\alpha = 3$, $\beta = 5$, and $\gamma = 0.5$. When calculating text embeddings for neighbor search, we weighted words according to POS: 1.0 for the target POS (noun and verb respectively), 0.1 for prepositions, 0.5 for other POS. But for calculating scores (i.e. s) we used uniform word weights.

Our algorithm turned out to be inefficient on nouns, with the submitted version scoring only 41.78% MAP on the private test set[7]. This is a little below the fastText baseline provided by the competition team near the deadline date. However, on verbs, our approach was more efficient and scored 44.83% MAP on the private test set, which is the best result so far.

5.1. Ablation study

In this section, we analyze the importance of different design decisions we made. The preliminary experiments were not well structured, so instead, we do an ablation study and show the effect of modifying some of our decisions. We evaluate MAP for nouns and verbs on our dev set and on the public test set. We do not report the MRR score, but its behavior is qualitatively similar to that of MAP. The results are summarized in Table 1.

From the table, we see that the model that we submitted performed worse than the baseline and the models of other participants on the public test set of nouns, but much better than the baseline and better than the competitors on the public test set of verbs. These results are consistent with the private test set.

We also see that some of the modifications to our model improve the MAP on a few datasets, but none of them improve the scores consistently on all the datasets.

Table 1: MAP of modified versions of the model

Model	nouns dev	nouns test	verbs dev	verbs test
Our submitted model	.4695	.4083	.2527	.4033
The best model of the competitors	—	.5590	—	.4032
The FastText baseline	—	.4343	—	.2760
$k = 30$	.4570	.3871	.2407	.3937
$k = 300$	.4561	.3983	.2664	.3884
$\alpha = 1$	.4699	.4084	.2573	.3987
$\alpha = 0$	.4216	.4093	.2587	.3909

[7] The leaderboard is available at https://competitions.codalab.org/competitions/22168#results

Model	nouns dev	nouns test	verbs dev	verbs test
$\beta = 1$	.4415	.4083	.2514	.4023
$\beta = 0$	.4216	.3639	.2466	.3799
$\gamma = 1$	.4396	.3963	.2429	.3677
$\gamma = 0$	.4753	.3857	.2587	.4016
FastText embeddings	.4263	.2432	.2237	.2615
s without POS weights	.4660	.4065	.2585	.4077
KNN with POS weights	.4653	.4071	.2338	.3900
Reduced index	.4627	.4121	.2671	.3645

All three parts of the ranking formula turned out to be useful: setting α or β to 0 or γ to 1 (effectively disabling parts of the formula) made the MAP scores deteriorate. When we changed the POS weighting scheme, MAP decreased in most cases as well. Replacing word2vec embeddings with FastText embeddings trained on the same Taiga corpus led to dramatically deteriorating performance.

One more subtle distinction of the proposed algorithm from the baseline is that it uses different terms of the synset separately in the search index. To validate this decision, we created an alternative index, when all entries in a synset are concatenated together before calculating an embedding and including it in the KNN index. This modification led to a visible increase in the test score for nouns, but the score for verbs dropped dramatically, so we decided not to submit this version.

6. Analysis

In this section, we analyze why our rather naive approach for hypernym discovery works and what it lacks.

6.1. Collecting vs ranking candidates

We start by comparing the impact of the quality of collecting and ranking candidate hypernyms on the overall quality. For this purpose, we estimate MAP with an oracle ranker on the dev set and get 81.6% (vs 47%) on nouns and 70.7% (vs 25%) on verbs. In more intuitive terms, this corresponds to 79% and 70% recall for nouns and verbs, respectively. It might mean that poor ranking is more responsible for the low score than poor candidate collection because with perfect ranking the gap between our solution and ground truth decreases by more than half. Our hypothesis is reinforced by the fact that our competitors' system that got the highest score on nouns was using features from numerous data sources for ranking.

6.2. Error analysis

To further understand the upsides and downsides of our system, we manually inspect 200 samples from the dev set and label the errors of our system on them. The frequencies of these errors are given in Table 2. If our system made several errors on a sampe, we assume their equal contribution.

Table 2: Relative frequency of model errors

Error type	Nouns	Verbs
No errors	.44	.12
Domain heuristics	.20	.40
Too general predictions	.08	.14
Homonymy	.05	.17
Abstract concept	.11	.01
Compositionality	.04	.09
Domain knowledge	.08	.00
Inversion of valence	—	.04
Labeling error	.00	.03

The major causes of errors include:

- *domain heuristics*: extracted neighbors are semantically related to the query (from the same domain), but reflect a different concept. For example, some of the close neighbors for **ЗАРЯЖАНИЕ** (loading or charging) are **ПРИЦЕЛИВАНИЕ** (aiming) and **ЛАФЕТ** (gun carriage), because they often occur together in the context of guns.
- *homonymy*: the term conveys multiple meanings, and golden homonyms are provided for one meaning, but neighbors — for another. For example, for **ВЫГОРАНИЕ** (fading, burnout) the golden hypernym is **ОБЕСЦВЕТИТЬСЯ** (to lose color), but our system provided **ГОРЕТЬ** (to burn).
- *too general predictions*: predicted hypernyms are more abstract then needed. For example, for **ПЕРЕСТАВЛЯТЬ** (to change places) the golden hypernyms include **СТАВИТЬ** (to set place), but our system predicted a more general **ПЕРЕМЕСТИТЬ** (to move).
- *abstract concepts*: the queries and the golden hypernyms are quite abstract (this is especially true for properties or processes), but our system interprets only one specific context of their usage, and does it wrong. For example, the word **ПОРТАТИВНОСТЬ** (portability) is out-of-vocabulary, so our system makes an inference about it from the word **ПОРТАТИВНЫЙ** (portable), which is distributionally close to gadgets, although semantically it is more general. As a result, the model predicts false hypernyms, such as **ЭЛЕКТРОННОЕ ОБОРУДОВАНИЕ** (electronic equipment).
- *compositionality*: mean word embeddings poorly reflect the meaning of a multiword term, because they are not aware of the syntax. For example, for the term **ПРОГРЕВАНИЕ БОЛЬНОГО МЕСТА** (warming up a sore spot) some of the predicted candidates are **БОЛЬНОЙ ЧЕЛОВЕК** (ill person) and **МЕСТО В ПРОСТРАНСТВЕ** (place).
- *domain knowledge*: predicting a correct hypernym requires specific knowledge about the world which is difficult to extract from distributional semantics. For example, one of the hypernyms for **ИРЛАНДСКАЯ СТОЛИЦА** (Irish capital) is **ГРАФСТВО** (county), but our model seems to be unaware of it.

- *inversion of valence*: the model mixes up the verbs describing the same process for the subject and the object. For example, for **ЗАТЮКАТЬ** (to harass) the model wrongly predicts a hypernym **ПЕРЕЖИТЬ** (to experience).
- *labeling errors*: the golden hypernyms do not correspond to the generally accepted meanings of the query term. For example, for **НАБИВАТЬСЯ** (to be stuffed, or to foist) there is a golden hypernym **НАДОЕСТЬ** (to pester), which is semantically related to the second sense of the query but seems to be its sibling, not a hypernym.

6.3. Areas of improvement

The analysis above indicates that the limitations of our method are mostly due to the limitations of static word embeddings themselves: they do not utilize morphology and syntax of the queries and provide only a narrow way of understanding their semantics. A better model would take into account:

- word morphology — to better extrapolate meaning between related words;
- phrase structure — to correctly integrate the meanings of the head and the dependent words into the phrase embedding, and to resolve homonymy;
- definitions of terms from external sources — to reason more correctly about the meaning of the rare words, and to use domain-specific relations;
- the structure of the taxonomy itself — to filter out too general or too specific hypernyms.

7. Conclusion

In this paper, we introduce a simple baseline for hypernym prediction, based solely on fixed word embeddings and their similarity. Its distinctive features are a large number of retrieved neighbors, nonlinear distance-based candidate scoring, and heuristics for obtaining phrase embeddings. Despite its simplicity, our system got the best score for the verbs track, which may indicate that nobody knows a smart way of predicting hypernyms for Russian verbs. Further research could integrate our system with other techniques of hypernymy prediction, which is necessary to overcome the limitations of static word embeddings.

References

1. *Berend, G. et al.:* 300-sparsans at SemEval-2018 task 9: Hypernymy as interaction of sparse attributes. In: Proceedings of the 12th international workshop on semantic evaluation. pp. 928–934 Association for Computational Linguistics, New Orleans, Louisiana (2018).
2. *Bernier-Colborne, G., Barrière, C.:* CRIM at SemEval-2018 task 9: A hybrid approach to hypernym discovery. In: Proceedings of the 12th international workshop on semantic evaluation. pp. 725–731 Association for Computational Linguistics, New Orleans, Louisiana (2018).

3. *Biemann, C.:* Ontology learning from text: A survey of methods. In: LDV forum. pp. 75–93 (2005).
4. *Biemann, C. et al.:* A framework for enriching lexical semantic resources with distributional semantics. Natural Language Engineering. 24, 265–312 (2017).
5. *Bojanowski, P. et al.:* Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 5, 135–146 (2017).
6. *Camacho-Collados, J. et al.:* SemEval-2018 task 9: Hypernym discovery. In: Proceedings of the 12th international workshop on semantic evaluation. pp. 712–724 Association for Computational Linguistics, New Orleans, Louisiana (2018).
7. *Espinosa-Anke, L. et al.:* Supervised distributional hypernym discovery via domain adaptation. In: Proceedings of the 2016 conference on empirical methods in natural language processing. pp. 424–435 Association for Computational Linguistics, Austin, Texas (2016).
8. *Hearst, M. A.:* Automatic acquisition of hyponyms from large text corpora. In: COLING 1992 volume 2: The 15th International Conference on Computational Linguistics. (1992).
9. *Jurgens, D., Pilehvar, M. T.:* SemEval-2016 task 14: Semantic taxonomy enrichment. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). pp. 1092–1102 Association for Computational Linguistics, San Diego, California (2016).
10. *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).
11. *Loukachevitch, N.:* Thesauri in problems of information retrieval [tezaurusy v zadachah informatsionnogo poiska], (2010).
12. *Loukachevitch, N. V. et al.:* Creating Russian WordNet by Conversion. In: Proceedings of Conference on Computatilnal linguistics and Intellectual technologies Dialog-2016. pp. 405–415 (2016).
13. *Mikolov, T. et al.:* Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013).
14. *Miller, G. A.:* WordNet: An electronic lexical database. MIT press (1998).
15. *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).
16. *Schlichtkrull, M., Martínez Alonso, H.:* MSejrKu at SemEval-2016 task 14: Taxonomy enrichment by evidence ranking. In: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). pp. 1337–1341 Association for Computational Linguistics, San Diego, California (2016).
17. *Shavrina, T., Shapovalova, O.:* TO the methodology of corpus construction for machine learning: "TAIGA" syntax tree corpus and parser. In: Proceedings of the international conference Corpus linguistics–2017. St. Petersburg. pp. 78–84 (2017).