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

Moscow, June 17–20, 2020

WORD2VEC NOT DEAD: PREDICTING HYPERNYMS OF CO-HYPONYMS IS BETTER THAN READING DEFINITIONS

Arefyev N. V. (nick.arefyev@gmail.com)[1,2,3],
Fedoseev M. V. (maxim.fedoseev13@gmail.com)[1],
Kabanov A. V. (arshehremen@gmail.com)[1],
Zizov V. S. (vzs815@gmail.com)[1]

[1]Lomonosov Moscow State University
[2]Samsung R&D Institute Russia
[3]National Research University Higher School of Economics, Moscow, Russian Federation

Expert-built lexical resources are known to provide information of good quality for the cost of low coverage. This property limits their applicability in modern NLP applications. Building descriptions of lexical-semantic relations manually in sufficient volume requires a huge amount of qualified human labour. However, given some initial version of a taxonomy is already built, automatic or semi-automatic taxonomy enrichment systems can greatly reduce the required efforts. We propose and experiment with two approaches to taxonomy enrichment, one utilizing information from word definitions and another from word usages, and also a combination of them. The first method retrieves co-hyponyms for the target word from distributional semantic models (word2vec) or language models (XLM-R), then looks for hypernyms of co-hyponyms in the taxonomy. The second method tries to extract hypernyms directly from Wiktionary definitions.

The proposed methods were evaluated on the Dialogue-2020 shared task on taxonomy enrichment. We found that predicting hypernyms of co-hyponyms achieves better results in this task. The combination of both methods improves results further and is among 3 best-performing systems for verbs. An important part of the work is detailed qualitative and error analysis of the proposed methods, which provide interesting observations of their behaviour and ideas for the future work.

Key words: lexical-semantic relations, hypernymy prediction, taxonomy enrichment, distributional similarity, definition extraction, word2vec, neural language models, XLM-R

DOI: 10.28995/2075-7182-2020-19-13-32

WORD2VEC ЖИВ: ПРЕДСКАЗЫВАТЬ ГИПЕРОНИМЫ КОГИПОНИМОВ — ЛУЧШЕ, ЧЕМ ЧИТАТЬ ОПРЕДЕЛЕНИЯ

Арефьев Н. В. (nick.arefyev@gmail.com)[1, 2, 3],
Федосеев М. В. (maxim.fedoseev13@gmail.com)[1],
Кабанов А. В. (arshehremen@gmail.com)[1],
Зизов В. С. (vzs815@gmail.com)[1]

[1]Московский Государственный Университет им. М. В. Ломоносова
[2]Московский Исследовательский Центр Самсунг
[3]Национальный исследовательский университет «Высшая школа экономики», Москва, Россия

Известно, что созданные экспертами лексические ресурсы предоставляют информацию высокого качества, но страдают низкой полнотой. Данная проблема влияет на их применимость в современных приложениях обработки текстов на естественных языках. Описание лексико-семантических отношений в достаточном объеме требует серьезных трудозатрат. При наличии уже сформированной начальной версии таксономии системы обогащения могут существенно сократить трудоемкость задачи. Мы предлагаем и сравниваем два подхода к обогащению таксономии, один из которых использует информацию из определений слов, а второй — информацию о контекстах, в которых слова встречаются, а также комбинируем эти подходы. Первый метод ищет когипонимы отсутствующего в таксономии слова либо с помощью дистрибутивных векторных представлений слов (word2vec), либо с помощью языковых моделей (XLM-R), а затем ищет гиперонимы найденных когипонимов в уже имеющейся таксономии. Второй метод старается извлечь гиперонимы непосредственно из определений Wiktionary.

Предложенные методы были оценены в рамках соревнования по обогащению таксономии на конференции Dialogue-2020. Мы показали, что предсказание гиперонимов когипонимов позволяет достичь более высоких результатов. Комбинация обоих методов привела к дополнительному улучшению результатов и вошла в тройку лучших систем для глаголов. Важной частью работы являются детальный качественный анализ и анализ ошибок предложенных методов, которые позволили сделать ряд интересных наблюдений и сформулировать идеи для дальнейшейших исследований.

Ключевые слова: лексико-семантические отношения, предсказание гиперонимов, обогащение таксономии, дистрибутивная близость слов, извлечение определений, word2vec, нейронные языковые модели, XLM-R

1. Introduction

Structured resources describing lexical-semantic relations between words such as synonymy, hypernymy, meronymy, etc. are known to require a lot of time and expert labour to build. However, modern machine learning methods allow to approximately predict which relations are held between words given unstructured resources like word definitions or word usage examples. Dialogue-2020 shared task on taxonomy enrichment [10] is a competition to compare these methods. The task is to predict possible positions of a word absent in a taxonomy. More technically, for each word in a test set 10 possible synsets of a given taxonomy shall be predicted, which are either direct hypernyms or hypernyms of hypernyms of this word.

We propose two approaches to the task, named Hypernyms-of-Co-Hyponyms (HCH) and Definition Processing (DP). HCH method tries to predict co-hyponyms of the word in question using a distributional semantic model (word2vec) or a language model (XLM-R), then maps them to the taxonomy and returns their hypernyms as the result. DP method extracts hypernyms from word definitions retrieved from Wiktionary. While HCH significantly outperform DP, the best results are obtained using their combination. Our combined system was the 3rd best performing system for verbs and the 5th for nouns.

2. Related work

Since this is the participating system description paper, we will describe only those work, which our methods are based on. Please, refer to [10] for an overview of the field.

To retrieve co-hyponyms of a given word, our HCH method exploits word2vec distributional semantic model [9] or XLM-R cross-lingual language model [3]. Word2vec learns low-dimensional vector representations of words (word embeddings), that are useful to predict their contexts in unlabeled corpora. Empirically, simple algebraic operations with learnt embeddings allow modeling some aspects of corresponding words' meaning. Most importantly for our work, using cosine similarity between word embeddings allows retrieving words, which are frequently co-hyponyms of a given word. In [2] the authors trained word2vec model for the Russian language on 150GB lib.rus.ec corpora and showed that this model outperform several other methods for retrieval of semantically similar words. We employed their word embeddings for the first step of our HCH method. XLM-R is a masked language model (MLM) trained on texts from 100 languages similarly to multilingual BERT [4], but having 3x more parameters, 2x larger vocabulary and trained not only on Wikipedias, but also Common Crawl, which increased training data for low-resourced languages by orders of magnitude resulting in more than 2TB of data in total. As a MLM, it is good at predicting words that can appear in a particular position in a given context. We exploit this ability to find words, that can either replace or appear in coordination with the target word in many different contexts.

A method very similar to our DP method was proposed as one of the baselines for SemEval-2016 Task 14: Semantic Taxonomy Enrichment [5]. In this task systems had to attach a new word into an existing taxonomy. Unlike our task, not only words but also

their definitions were given as input to the systems. First Word First Sense approach was a very strong baseline, which was outperformed by only one participant. This baseline looked for the first word in the definition with the same part of speech as the given word. Then the corresponding synset was assumed to be the correct hypernym. Extending this approach, our DP method first retrieves definitions and then extracts not only words, but also phrases that can represent hypernyms of the word defined. This modification is required since in our task most synsets are represented by phrases.

3. Taxonomy enrichment methods

3.1. Hypernyms-of-Co-Hyponyms (HCH)

The Hypernyms-of-Co-Hyponyms (HCH) approach is based on the assumption, that the words distributionally similar to the target word are often its co-hyponyms, hence they have the same hypernyms we would like to predict (see **Appendix B** for detailed analysis regarding this assumption). As the simplest default option, we employ Skip-Gram Negative Sampling word2vec model [9] trained on 150GB lib.rus.ec corpus of books mostly in Russian [2] with cosine similarity metric to find k nearest neighbours for the target. This option works bad for words that are either absent or very rare in lib.rus.ec, thus, do not have good embeddings (**Appendix C** explores the correlation between word frequencies and hypernyms prediction quality). Nevertheless, we found it to be the best performing option.

Alternatively, we can find occurrences of the target word in some corpus, retrieve their contexts and ask a language model which words can replace or stand in co-ordinated row with the target in these contexts. Specifically, for each target word we have retrieved examples from the news corpora [10] using Sphinx search engine[1] and passed them to XLM-R [3]. The target can be replaced with the special token <mask>, so the model will receive the same kind of input it was trained on. However, this hides the target from the model and often results in predictions, that are plausible but entirely unrelated to the target. To stimulate predicting co-hyponyms we employ dynamic patterns proposed by [1], i.e. replace the target with a pattern like **"<mask> and T"** and then replace **T** back with the target. Thus, instead of *I love <mask>*, the model receives *I love <mask> and cats*. For each example we take 100 tokens that are most probable in the masked position as substitutes for the target in this example (this number is selected intuitively, selecting it as a hyperparameter may improve results). For each target we take k most frequent substitutes across all examples as the nearest neighbours. Since XLM-R has a vocabulary of 250K subwords shared for 100 languages, it contains only a small number of frequent Russian words. We found it beneficial to predict substitutes consisting of two subwords by inserting two masks, taking 100 most probable predictions for the first one and then one most probable continuation for each of them. We leave exploration of other multitoken substitutes generation techniques for the future work.

[1] http://sphinxsearch.com/

For each nearest neighbour we find all matching synsets in RuWordNet [7]. Exact matching (i.e. retrieving only synsets having exactly the same word as one of their expressions) resulted in better final performance than inexact matching with limited Levenshtein distance. For XLM-R we additionally performed lemmatization. Then for each matched synset we find its hypernyms. Finally, for each target word we return 10 most frequent hypernyms of its nearest neighbours as the result.

For a few target words, **table 1** shows their nearest neighbours, that were matched to some synsets. Nearest neighbours having direct or second level hypernyms that are also correct hypernyms of the target word, or nearest neighbours that are correct hypernyms themselves, are in bold. These nearest neighbours contribute towards correct predictions. See **section 4.1** and **appendix A** for additional analysis and discussion.

Table 1: Target words with their top-15 nearest neighbours, that were found in the taxonomy. Neighbours that resulted in correct hypernyms predicted are in **bold**

Target word	word2vec	XLM-R (<mask><mask> or T)
переиздавать	**переиздать**, **публиковать**, **издаваться**, рецензировать, **перепечатывать**, **печататься**, **печатать**, переиздание, **опубликовать**, издание, **напечатать**, двухтомник, переписывать, допечатывать, перечитывать, ...	исполнять, переписывать, выдавали, повторили, дополнять, переводить, **издавать**, повторять, **издали**, **издавали**, переписали, ставил, повторял, показывал, **издавая**, ...
пылать	**полыхать**, **гореть**, **пламенеть**, **сгорать**, **запылать**, **тлеть**, **сиять**, кипеть, **разгораться**, **трепетать**, **загораться**, дымиться, **вспыхивать**, **сверкать**, **загореться**, ...	**горят**, падает, **горит**, спала, **горела**, трус, **горящего**, **горящих**, пламя, **тает**, гаснет, шатается, погибает, тонет, жрет, ...
прожевывать	**пережевывать**, **прожевать**, **жевать**, откусывать, **разжевать**, **проглатывать**, **заглатывать**, **пережевать**, **глотать**, **съедать**, наедаться, **дожевать**, **съесть**, **проглотить**, **запивать**, ...	солить, чистить, **кушать**, жуть, **жрать**, **пробовать**, пропускать, варить, **дробить**, растворять, резать, мять, **жевать**, смывать, удалять, ...
первомай	первомайский, субботник, **праздник**, новогодний, юбилей, **праздничный**, годовщина, праздновать, отмечать, предпраздничный, пятидесятилетие, митинг, парад, именины, летие, ...	**первого мая**, **Праздник**, рабочий день, **Первомай**, **первое мая**, Мая, Независимость, апрель, выходной, Великий, **Новый год**, ВОВ, рабочего дня, **праздника**, **День труда**, ...
атлетизм	спортивность, атлетичность, **спорт**, **бодибилдинг**, **культуризм**, выносливость, атлетический, олимпизм, мужественность, культуристский, мотоспорт, артистизм, техничность, мускулистость, физкультура, ...	**гольф**, ловкость, кросс, скоростью, сильным, умом, умением, активностью, фитнеса, **бокса**, **гимнастики**, секции, ретро, конь, чутье, ...

POS mapping. While the correct hypernyms always have the same part of speech (POS) as the target, nearest neighbours often contain words with different POS, resulting in incorrect predictions. Predictions with incorrect POS can be simply filtered. However, we noticed that derivationally related words with different POS usually correspond to synsets with identifiers differing only by the last letter. For instance, the synset '144051-**N**' consists of words: издательское дело (publishing), печатание (printing), напечатание (typing), издание (publication), and '144051-**V**': выходить из печати (see the light), печатать (type), напечатать (print), издать (to issue), издаваться (to be published). Changing POS in the synset identifier, i.e. replacing the letter with the desired, often results in additional correct predictions. Therefore, after finding the nearest neighbours and their synsets, for synsets with the wrong POS we replace their POS with the POS of the target. Only if the resulting synset is not found in the taxonomy, we remove it from predictions.

Hypernyms ordering and merging. We use not only direct hypernyms of nearest neighbours (denoted as *degree 1*), but also hypernyms of hypernyms (*degree 2*), they are compared in **section 4.1**. Given a list of nearest neighbours of a particular target, we build two ordered dictionaries, for their direct and indirect hypernyms separately, with hypernyms as keys and their counts as values. Hypernyms appear in the decreasing order of their counts, and if the counts are equal, in the order of corresponding nearest neighbours. Thus, if there are several nearest neighbours with the same hypernym, this hypernym will be among the first hypernyms in the dictionary and will more likely be returned as the result. Also, if there are several hypernyms with the same counts, we will return those corresponding to more similar neighbours first. The best results are achieved by merging direct and indirect hypernyms (*degree 1 + 2*). Counts of each synset occurred as a direct and indirect hypernym of nearest neighbours are added. For hypernyms with the same counts the order of corresponding nearest neighbours are preserved. Thus, we return 10 most frequent hypernyms of nearest neighbours, preferring hypernyms of neighbours that are more similar to the target word if the counts are equal.

3.2. Definition processing (DP)

The definition processing method extracts hypernyms from Wiktionary definitions. Wiktionary is a lexical semantic resource, introduced in [11]. It contains word definitions, examples of word usage, and some metainformation. Definitions can be classified into two large groups, *intensional* definitions try to give the sense of a term and *extensional* definitions try to impose the objects that a term describes [8]. We are expecting to see the *intensional* ones. The simplified version of the DP method assumes that some phrase in the definition is the hypernym or at least reflects some connection with the hypernym of the word defined, and this hypernyms is already described in the taxonomy. Thus, for each N it looks at all the N-grams of the definition D and finds out which have corresponding synsets $S(N, D)$ in the taxonomy. We fix the maximum N for which $S(N, D)$ is not empty. Such N-grams and their corresponding synsets are considered to be hypernyms. For example, the phrase 'A B C' will be processed from N = 3 to N = 1. First, if the taxonomy has synset 'A B C', it will be the only answer. Next, the method will search 'A B' and 'B C' in the taxonomy and return those

that were found. If none of then was found, the same procedure will be performed for 'A', 'B', 'C' separately.

After analysing errors of this simplified method, we found situations when the extracted hypernym is not found in the original taxonomy. This problem was solved by recursively enriching of the taxonomy. In the first iteration we extract from each definition D all the N-grams with $N = 0.8 * len(D)$ and map them to the taxonomy. For definitions with non-empty $S(N, D)$ we add the defined terms to the taxonomy and specify $S(N, D)$ as their hypernyms. Then we exclude these definitions from the iterative process. We decrease N for each definition by one on each iteration and repeat hypernyms search and taxonomy enrichment procedure. Thus, terms with short definitions are added to the taxonomy first. For short definitions, the precision of each iteration (proportions of true positives in $S(N, D)$) is larger than for long definitions. The intuition is that in initial iterations we introduce less noise to the taxonomy, and that is why the noise grows more slowly in the later iterations. Thanks to such step-by-step filling in, we can detect, for example, the relation "айпад — планшетный компьютер" ("ipad" IsA "tablet computer"). This happens in the following way. There are two different definitions: *Планшет — то же, что планшетный компьютер...* (*a tablet is the same as a tablet computer...*), and *айпад — планшет марки iPad* (*ipad is an iPad brand tablet*). First, a relation: "планшет — планшетный компьютер" ("tablet" IsA "tablet computer") will be built, and then "айпад — планшет" ("ipad" IsA "tablet").

4. Experiments

To develop our models and select their hyperparameters during the evaluation period, instead of using the development set provided by the organizers, we have decided to build another development set that is more similar to the test set. To achieve this, the words from the public test set were sorted by their frequencies in the news corpus, and then divided into 10 bins containing equal number of words. All the words from the given (train) part of the taxonomy were mapped to the same bins. We have sampled 50 train words from each bin. This resulted in the development set, that consists of 500 words having the same frequency distribution as the test words. However, other characteristics of the obtained development set may still be far from the test set. Hence, in during post-evaluation period we have decided to perform hyperparameter and error analysis on the public test set, leaving the private test set for the final evaluation of our methods performance.

4.1. Hypernyms-of-Co-Hyponyms

Figure 1 shows the dependence of the results of HCH method using word2vec from hyperparameters on the public test set. MAP is low if we use only the synsets of nearest neighbours (*degree 0*). This is intuitive because only few nearest neighbours are hypernyms of the target word. It is better to use direct hypernyms (*degree 1*), than indirect ones (*degree 2*). The combinations of hypernyms perform significantly better. Also, we see that larger top-k perform better. We pre-calculated only 300 nearest neighbours, however, increasing this number will likely increase the results further.

Table 2 compares HCH method using word2vec (default) and XLM-R to retrieve nearest neighbours. Evidently, word2vec results in better nearest neighbours for the task. Regarding the XLM-R model, from figure 2 and table 2 we see that two-subword substitutes perform much better, and **"<mask><mask> or T"** template is consistently better than **"<mask><mask> and T"**. Additionally, POS mapping gives small but consistent improvements for word2vec.

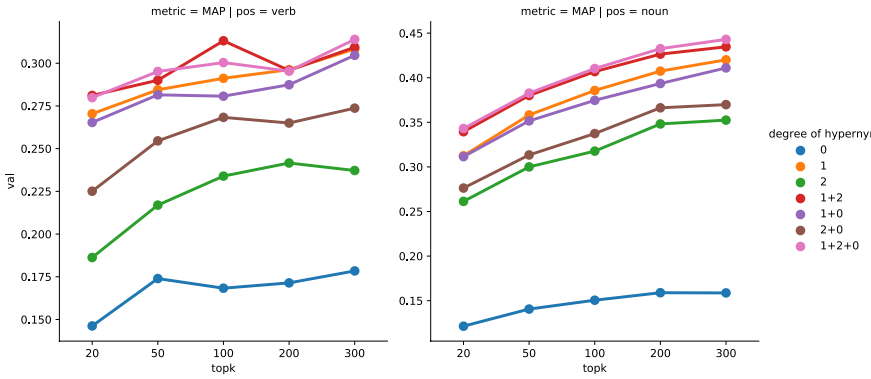

Figure 1: Evaluation of HCH on the public test set

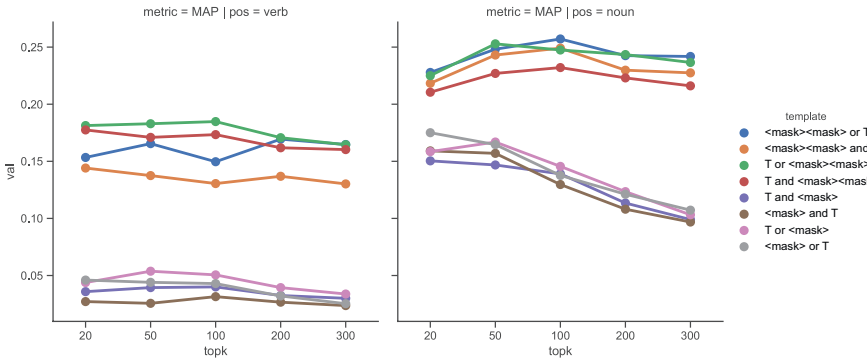

Figure 2: Evaluation of HCH with XLM-R on the public test set with degree of hypernyms 1+2

The detailed error analysis of HCH method can be found in appendix D. From examples in table 1 and appendix A, it seems that word2vec produces less frequent and more diverse words, which are closely-related to the target, however, they are not exclusively co-hyponyms, but also topically related words. The error analysis revealed that such nearest neighbours is the largest source of errors. XLM-R generates more distantly related (and sometimes seemingly unrelated) words. One possible reason is generation of substitutes consisting of one or two subwords only, which greatly limits possible substitutes. Another possible reason is that XLM-R is trained with cross-entropy

loss resulting in prediction of frequent words first, while word2vec is trained with negative sampling loss, which promotes words having high PPMI with the target [6]. The potential advantage of XLM-R is its ability to generate not only words, but also phrases (notice *New Year* among other holidays in **table 1**). See **appendix B** for additional study of relations between targets and their nearest neighbours.

Rare words may have bad word2vec embeddings or no embeddings at all, resulting in inadequate nearest neighbours and bad performance. **Appendix C** investigates this problem. It shows that the performance of our method drops for the target words having less than 100–200 occurrences in lib.rus.ec. Luckily, the majority of words have more occurrences, so this is not the major problem.

4.2. Definition Processing

Table 2 shows that pure DP method is much worse than HCH method using word2vec. It is comparable with the results of HCH using XLM-R. Detailed error analysis can be found in the **appendix E**, while this section summarizes problems discovered.

Most errors (about 43%) are words in the test set, that do not have definitions in Wiktionary. Another problem is using exact matching to find synsets corresponding to the extracted N-grams. For example, phrases “ambulance station” and “ambulance substation” are considered different. These mistakes constitute about 10% of errors.

Table 2: Model comparison on the public test set. DP^k means DP with k iterations. $HCH^{topk,\ degrees}$ means HCH with top-k neighbours and specified degrees of hypernyms

Method	Nouns MAP	Nouns MRR	Verbs MAP	Verbs MRR
$HCH^{300,\ 1+2+0}$	**0.4430**	**0.4728**	0.3130	0.3531
$HCH^{300,\ 1+2+0}$ w/o POSmap	0.4341	0.4615	0.3061	0.3472
XLM-R (<mask><mask> or T)	0.2480	0.2766	0.1844	0.2121
XLM-R (<mask><mask> and T)	0.2338	0.2659	0.1472	0.1723
XLM-R (<mask> or T)	0.1673	0.1926	0.0543	0.0641
XLM-R (T and <mask>)	0.1539	0.1786	0.0377	0.0410
DP^{30}	0.1894	0.2175	0.1904	0.2519
HCH^{50} + DP^3	0.4165	0.4538	0.3122	0.3613
HCH^{50}+ DP^{12}	0.4045	0.4365	**0.3586**	**0.4072**
HCH^{100}+ DP^{12}	0.4196	0.4540	0.3548	0.4019

Some predictions also have a very low precision, i.e. small proportion of true positives in the predicted answers. Meanwhile, we can identify which of the answers are potentially correct. Particularly, this can be obtained from the fact that sentences are usually built in a sufficiently simple way. The corresponding definition was constructed on average as a single sentence with no more than two objects. It implies that one, two, or three synsets can be found in the definition. As an example, if we have target word *пустельга* (*kestrel*) and DP method returned *хищная птица, легкомысленный человек* (*predatory bird, frivolous person*) (N = 2), we consider them to be

right answers. But if we have *книга, науки, прекратить, исключительно, знакомить, потом* (*book, sciences, stop, exclusively, meet, after*) (N = 1) for target word *сонник* (*dream book*), it is the signal that our answer is possibly wrong. Therefore, it was able to see that if we found more than three hypernym synsets for the target word, the output is more likely to be polluted. Such results were rejected, and did not contaminate taxonomy (8%). Moreover, if N was reduced to 1, some individual irrelevant words (auxiliary verbs, as a common example) could appear in the output. At N > 1 most of the extracted senses were true hypernyms. The disadvantage of definitions was in a fact that the position in the taxonomy could differ by one or several relation steps from the answer. The problem with precision was solved by intersecting predictions with the HCH predictions. The completeness problem still remains.

5. Final combined model

The combined method used voting. Hypernyms predicted by both HCH and DP methods were returned first, their order in HCH answers was preserved. If the HCH returned less than 10 hypernyms, then the output was supplemented by the answers of the DP method. For instance, if HCH predicted 'A B C D' and DP predicted 'C B E', then the combined model returned 'B C A D E'.

The results of the combined model are shown in table 2. Interestingly, iteration DP method improves quality for verbs, but worsens it for nouns. In general, DP method makes HCH results more accurate, but has significant limitations if used alone.

6. Conclusion

In this paper we have proposed and evaluated two methods of taxonomy enrichment, exploiting information both from word definitions and examples of word usage. We have shown that their combination improve both of them, resulting in 3rd best result for verbs and 5th for nouns (table 3) on Dialogue-2020 taxonomy enrichment shared task.

Table 3: Final results on the private test set

Competitor	Nouns MAP	Nouns MRR	Verbs MAP	Verbs MRR
Our submissions: team *MorphoBabushka*				
$HCH^{300,\ 1+2} + DP^{30}$	0.4497 (5)	0.4835 (5)	0.3890 (3)	0.4419 (3)
DP^{30}	0.1729 (13)	0.1931 (13)	0.1399 (11)	0.1690 (11)
Top participants' submissions				
cointegrated	0.4178 (6)	0.4503 (6)	**0.4483** (1)	0.5049 (2)
Yuriy	**0.5522** (1)	**0.5940** (1)	0.4355 (2)	**0.5130** (1)
xeno	0.5054 (2)	0.5433 (2)	0.3075 (4)	0.3547 (4)
KuKuPl team	0.4976 (3)	0.5332 (3)	0.2470 (6)	0.2897 (6)
RefalMachine	0.4930 (4)	0.5314 (4)	0.2542 (5)	0.2969 (5)

Though it proved to be hard to apply modern language models like XLM-R to the task of taxonomy enrichment compared to the simplicity of word2vec, we believe

that further work on substitute generation using such models can result in new SOTA in this task. At the same time, using dictionaries or similar resources to solve the task seems to give bad results unless combined with other strong methods. These resources contain only a small fraction of the target words and the definitions are usually misleading. Still it worth trying to extract good definitions from the whole Web using search engines and definition extraction techniques.

7. Acknowledgements

We thank the organisers of the competition for such an inspiring task. We are grateful to our reviewers who motivated us to do detailed qualitative and error analysis of our methods, which hopefully make our paper more interesting to the readers. The contribution of Nikolay Arefyev to the paper was partially done within the framework of the HSE University Basic Research Program funded by the Russian Academic Excellence Project '5-100'.

Appendix A. Examples of nearest neighbours found in the taxonomy

For a few target nouns and verbs, table 5 shows nearest neighbours (15 at most) from top-300 word2vec and top-100 XLM-R nearest neighbours that were matched to some synsets. Nearest neighbours that resulted in correct predictions (i.e. nearest neighbours having direct or second level hypernyms, that are also correct hypernyms of the target word, or nearest neighbours that are correct hypernyms themselves), are in bold. Words "обгорать" and "пропихнуться" don't have nearest neighbours in XLM-R because in the news corpus there are no examples with these words, there are only with the words "обгораться" and "пропихнуть".

Table 4: Distribution of counts of the nearest neighbours found in the taxonomy among top-100 nearest neighbours

	verbs				nouns			
quantile	word2vec	word2vec + lemm	XLM-R	XLM-R + lemm	word2vec	word2vec + lemm	XLM-R	XLM-R + lemm
0.05	1	9	0	57	0	5	4	31
0.25	7	27	3	75	4	16	11	58
0.50	16	46	32	84	9	29	25	75
0.75	26	62	67	90	15	42	39	85
0.95	43	78	85	96	39	63	61	94

Table 4 shows distribution of counts of the matched nearest neighbours among top-100 nearest neighbours. Obviously, lemmatization increases the number of matched neighbours for both models. However, our experiments have shown that for word2vec it does not improve final results, so we did lemmatization only for XLM-R. This is due to the fact that given a word in its base form, word2vec often returns similar words

in their base forms also. Unlike word2vec, XLM-R is a language model and returns those forms that are appropriate given particular contexts. Also, there more matched neighbours for XLM-R than those for word2vec, probably because XLM-R generates only 2 subword units, which are mostly frequent words and phrases, hence, they are more likely described in the taxonomy.

Table 5: Nearest neighbours that were found in the taxonomy. Neighbours that resulted in correct hypernyms predicted are in **bold**

target word	word2vec	XLM-R (<mask><mask> or T)
варево	**похлебка**, пойло, **суп**, **рагу**, помешивать, зелье, котелок, **супчик**, **кушанье**, котел, снадобье, половник, хлебать, размешивать, **бульон**, ...	квас, варка, мясо, сало, смесь, картофель, соус, зелень, ведро, **супа**, соль, творог, специи, еда, медь, ...
доктор	**профессор**, **врач**, **психиатр**, **медик**, **хирург**, коллега, **ассистент**, невропатолог, пациент	**профессора**, **мастера**, **докторант**, **врачей**, **психолога**, **хирурга**, специалиста, ученого, бывшего, ученый, академии, адвоката, юриста, **студента**, журналиста, ...
радиожурналист	**тележурналист**, **журналист**, **радиоведущий**, **телеведущий**, **телеобозреватель**, эссеист, **колумнист**, прозаик, **обозреватель**, **корреспондент**, международник, фотожурналист, литератор, **публицист**, телепродюсер, ...	авторов, **журналистов**, экспертов, редакторов, **репортеров**, ведущих, продюсеров, программистов, операторов, исполнителей, агентов, выпускников, писателей, фотографов, режиссеров, ...
праправнучка	**правнучка**, **внучка**, **родственница**, **племянница**, **прабабка**, **прабабушка**, **правнук**, ровесница, **внук**, **дочь**, **кузина**, **праправнук**, **сестра**, **потомок**, **наследница**, ...	**внука**, **бабушка**, **дочка**, **наследница**, вдова, **наследника**, няня, сынок, **девица**, **супруга**, **невеста**, **подруга**, **сестра**, щенка, родная, ...
задушевность	проникновенность, **сердечность**, музыкальность, **душевность**, **непринужденность**, доверительность, ласковость, мелодичность, **искренность**, **непосредственность**, трогательность, трепетность, выразительность, приветливость, **естественность**, ...	**душевность**, легкость, живость, **искренность**, **откровенность**, **непосредственность**, страсть, романтика, торжественность, чувственность, тонкость, эмоциональность, ярость, холодность, мягкость, ...
переплести	**переплетать**, **сплести**, **заплести**, вплести, расплести, переплетенный, **переписать**, **перепечатать**, **отпечатать**, скрепить, сплестись, вклеить, сцепить, обмотать, **напечатать**, ...	**связать**, разделить, развязать, **совмести**, **связывать**, синхронизировать, **увязать**, **перевязать**, **сочетать**, привязать, завязать, **смешать**, **вязать**, **свести**, **объединять**, ...
отвертеть	**открутить**, **отвинтить**, отвинтиться, развертеть, откручиваться, накутывать	*no nearest neighbours*

target word	word2vec	XLM-R (<mask><mask> or T)
обгорать	**обгореть**, облезать, шелушиться, **обугливаться**, **выгорать**, **смуглеть**, отслаиваться, **обуглиться**, **чернеть**, дымиться, поджариваться, **загореть**, плавиться, трескаться, **посмуглеть**, ...	падает, распадается, разводится, **горит**, выпадает, красится, **белеет**, попадает, страдает, тонет, шатается, стареет, цветет, теряется, портится, ...
пропих-нуться	**протолкнуться**, **протиснуться**, **протолкаться**, втиснуться, продохнуть, пролезть, **протискаться**, давка, **протискиваться**, **пропихиваться**, впихнуть, пропихнуть, затолкать, **проталкиваться**, запихнуть, ...	*no nearest neighbours*
утаскивать	**утащить**, затаскивать, **оттаскивать**, **увозить**, **уводить**, забирать, выволакивать, заманивать, притаскивать, **похищать**, подтаскивать, **вытаскивать**, **тащить**, **таскать**, **волочь**, ...	**брали**, **берут**, везут, несут, переносили, **ведут**, **водили**, рвет, несли, **выносили**, носили, забрали, **убирали**, поднимали, **бросали**, ...

Appendix B. Relations between target words and their nearest neighbours

Table 6 shows the nearest neighbours, that were found in the taxonomy (10 at most), and their relation to the target. Only synsets that have a common hypernym (direct or transitive) are considered related. CH^{X+Y-} stands for co-hyponyms, which are connected to the target by a path of X upward and Y downward edges, *hypoX* and *hyperX* stand for hyponyms and hypernyms of level X and *syn* are synonyms of the target. *none* means that the target and its nearest neighbour do not have any common hypernym at any level.

Table 6: Examples of relations between target words and nearest neighbours

target word	word2vec	XLM-R
балкарка	карачаевка(CH^{1+1-}), кизилюрт(CH^{5+5-}), адыгейка(CH^{1+1-})	татарка(CH^{1+1-}), украинка(CH^{1+2-}), чешка(CH^{1+2-}), грузинка(CH^{2+2-}), чувашка(CH^{1+1-}), арабка(CH^{2+1-}), русалка(CH^{5+3-}), киргизка(CH^{1+1-}), монголка(CH^{2+2-}), северянка(CH^{5+3-})

target word	word2vec	XLM-R
узость	ограниченность(syn), односторонность(hypo1), поверхностность(CH^{2+1-}), неразвитость(CH^{1+1-}), замкнутость(CH^{3+4-}), однобокость(hypo1), узкость(syn), примитивность(CH^{2+3-}), убогость(CH^{1+1-}), мелочность(CH^{3+3-})	слабость(CH^{1+1-}), недостаточность(hyper1), ограниченность(syn), грубость(none), незнание(CH^{4+2-}), закрытость(CH^{3+2-}), плотность(CH^{2+2-}), неопределенность(CH^{3+3-}), низость(CH^{3+4-}), особенность(CH^{3+1-})
интеллект	интеллектуальный(none), разум(syn), мышление(CH^{2+1-}), потенциал(CH^{4+4-}), интеллектуальность(syn), ум(syn), мозг(CH^{6+5-}), умственный(none), сообразительность(CH^{3+1-})	нейрон(CH^{6+4-}), НИИ(CH^{6+5-}), интуиция(none), совесть(CH^{5+3-}), смех(none), воспитание(none), мусор(CH^{6+4-}), ученый(none), геном(CH^{6+4-}), искусственный(none)
восставать	восстать(syn), бунтовать(syn), покоряться(CH^{5+6-}), отрекаться(CH^{3+3-}), ополчаться(CH^{2+2-}), роптать(CH^{1+1-}), ниспровергать(CH^{4+5-}), негодовать(CH^{5+3-}), ополчиться(CH^{2+2-}), возмущать(CH^{4+4-})	становиться(none)
застег-нуться	застегнуть(syn), расстегнуть(CH^{3+3-}), застегивать(syn), зашнуровать(CH^{1+3-}), запахнуть(CH^{2+2-}), одеться(CH^{5+3-}), застегиваться(syn), надеть(CH^{3+5-}), расстегнуться(CH^{3+3-}), раздеться(CH^{3+5-})	завязать(CH^{3+5-}), обуться(CH^{3+6-}), остановиться(none), запнуться(CH^{3+5-}), встать(none), одеться(CH^{5+3-}), убежать(CH^{3+5-}), связать(CH^{1+1-}), задеть(CH^{3+2-}), нажать(CH^{3+2-})

Figure 3 shows the proportion of nearest neighbours related by different relations to the target words among topk nearest neighbours. These proportions are estimated on the nouns from the public test set. Only nearest neighbours, that are *synonyms*, direct or second level hypernyms (*hypernym12*), direct hyponyms (*hyponym1*), or co-hyponyms connected to the target by 1 or 2 upward followed by 1 or 2 downward edges (*CoHypo12*) result in correct answers predicted. The proportion of the synonyms and suitable hypernyms shrink rapidly, when topk grows, because nearest neighbours are sorted by cosine similarity metric, and we take the less close vectors for the target word, increasing topk. Proportion of the *CoHypo12* practically does not shrink. XLM-R has much less synonym, hypernyms and *CoHypo12*, much more distantly related words, that's why it works worse than word2vec. The target words with wrong predictions have more distantly related (only by topic) or unrelated neighbours.

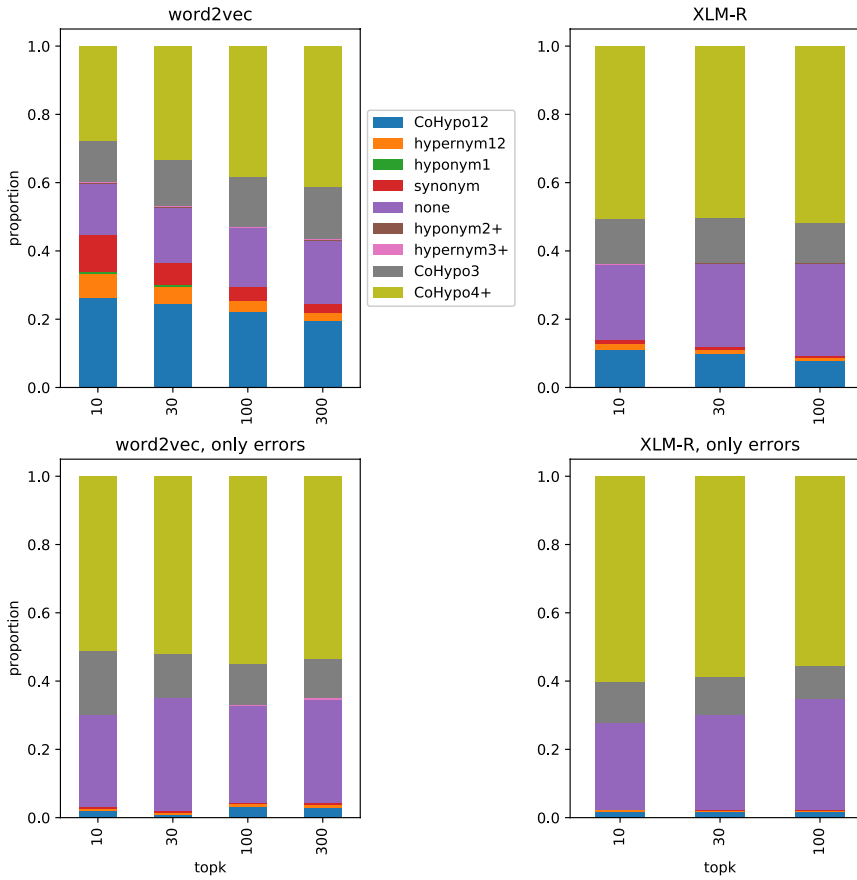

Figure 3: Proportions of nearest neighbours related by different relations to the target

Appendix C. Performance depending on the target word frequency

In this section we explore how the results of our HCH method vary depending on the frequency of the target word in the corpus that word2vec was pretrained on. Figure 4 shows MAP and MRR for the words from the public test set depending on their frequencies in lib.rus.ec. Approximately 7% of the words have no word2vec embeddings (because they occurred less than mincnt=5 times in lib.rus.ec). For them, we backoff to predicting 10 most frequent hypernyms as estimated on the development set. All target words that have word2vec embeddings were divided by their frequencies into 10 bins, containing equal number of words each. Hence, performance on each bin contribute equally to the final system performance. From the figure we conclude that for the words that occurred at least about 200 times or more there is no large dependence between their frequency and performance. For the words with

less than 100–200 occurrences the performance drops significantly. Finally, most frequent hypernyms backoff doesn't work at all.

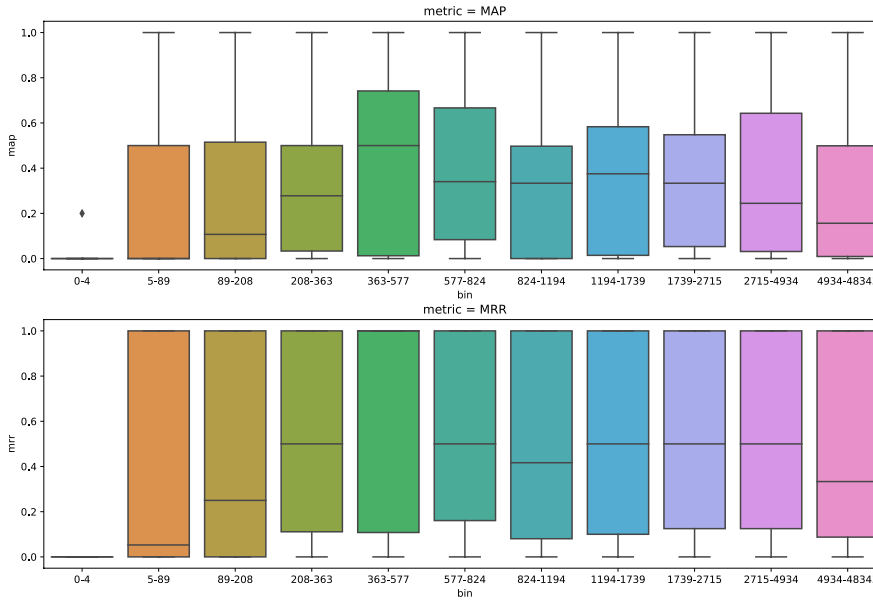

Figure 4: Dependence of the result of the word frequency

Appendix D. Error analysis of HCH method

Table 7 shows the results of error analysis of HCH method made for the public test set. In total, 100 randomly selected errors were examined, including 50 for XLM-R and 50 for word2vec. We divided all errors into two large groups: those examples which had correct answers among all HCH predictions, but not among top-10 final results (not-in-10), and those with no correct predictions at all (not-in-preds).

Related words. The most common type of errors are due to the nearest neighbours that are somehow related to the target word, but are not its co-hyponyms. For example, the word *обух* (butt/head) has correct hypernyms *сторона, бок, боковая часть* (side). Its substitutes are *топор* (axe), *тесак* (cleaver), *заточка* (sharpening), *рукоять* (handle), *клинок* (blade), which are not hyponyms of *боковая часть* (side). Potentially, this may be fixed by intersecting word2vec and XLM-R outputs. In the case of XLM-R itself, it is worth trying to experiment with other templates.

High-level hypernym. The second type of errors is related to the hierarchy of hypernyms. This means that the obtained answers of the system turned out to be either too abstract, or even the substitutes themselves are already very high-level hypernyms. An example is the word *саркома* (sarcoma) that has true answer *опухоль, злокачественное новообразование* (tumor) and predicted answers *болезнь* (illness),

воспаление (inflammation), *инфекционная болезнь* (infectious disease), *онкологическое заболевание* (cancer), *физическое самочувствие* (physical well-being). As another example, the word *космы* (tousle) has the true answer *прядь волос, пук, пучок* (lock of hair, topknot), while substitutes already contain a more abstract hypernym *волосы* (hair). At the same time, we have found no cases when true hypernyms are more high-level than the predicted ones. In the process of analyzing errors, it was noticed that for errors of this type the right answers nevertheless have a large number of occurrences, but not as large as those of higher-level hypernyms. Therefore, it makes sense to give higher priority to the predicted hypernyms with lower level of abstraction, rather than just the most frequent ones.

Table 7: Error types and counts

	XLM-R		word2vec	
Error type	**not-in-10**	**not-in-preds**	**not-in-10**	**not-in-preds**
Related words, not co-hyponyms:	12	6	6	17
High-level hypernym	13	2	3	6
incl. prevalence of higher-level hypernyms in the answers	11	1	3	5
incl. substitutes are already higher level hyponyms	2	1	0	1
Incompleteness of taxonomy	4	2	4	5
incl. synonymy	1	2	2	4
incl. minor meaning	3	0	2	1
Complex taxonomy	3	4	2	4
Incorrect interpretation of the word	1	3	2	1
incl. truncated words	1	0	2	1
incl. names	0	3	—	—
Total	**33**	**17**	**17**	**33**

Incompleteness of taxonomy. Some errors are not exactly errors, but cases where the system found correct hypernym that was not among gold answers. It can be a synonym of some gold answer. For example, the word *кейтеринг* (catering) has the gold answer *есть, принимать пищу* (eat, digest food), while the system predicts *питаться, кормиться* (feed). Sometimes the system predicts hypernyms of some secondary meanings of the target word. For example, *конторка* (check/bureau) has only gold hypernyms for one of its senses *письменный стол* (desk), but not the other sense *маленькая контора* (small office). Another good example is the word *аквабайк* (jetski). In the context of *Помимо дайвинга, здесь можно заняться виндсерфингом, аквабайком, кайтингом, полетать на парашюте и вертолете* (In addition to diving, you can do windsurfing, jetskiing, kiteboarding, parachuting and helicopters), the word *аквабайк* (jetski) is clearly used as a sporting event, while in the correct answers it appears only as a vehicle.

Complex taxonomy. This type of errors means that the system's answers are suitable, but very different from the correct answers. Examples are the words:

- *каторжник* (convict)—the correct answer is *лицо, отбывающее наказание* (person, sending a punishment), answers of the system is *преступник* (criminal), *осужденное лицо* (convicted person)
- *фианит* (cubic zirconia)—the correct answer is *кристалл, твердое тело, химическая продукция, синтетические материалы* (crystal, solid, chemical products, synthetic materials), answers of the system is *драгоценный камень* (gemstone), *минерал* (mineral), *корунд* (corundum), *природное минеральное образование* (natural mineral formation).

Incorrect interpretation of the word. There are errors associated with incorrect interpretation of the target word. For example, *стрип* (strip) has the correct answer *денежная ценная бумага, ценная бумага* (monetary securities, securities), while word2vec returns *стриптиз* (striptease) as its nearest neighbour. In the case of XLM-R, there are contexts where target words are used as person names or as a part of some titles. This can be fixed using named entity recognition to filter such contexts.

The performed error analysis revealed that the most common type of errors is due to the fact that the nearest neighbours are not exclusively co-hyponyms, but often topically related words. Hence, it may be worth working on better co-hyponyms prediction.

Appendix E. Error analysis of DP method

Table 8 shows the approximate percentage of errors, that were detected in a random sample (49 target words) from the public test.

In reality, some sentences have such a construction, that it is impossible to extract a hypernym from them. In addition to that, they can seek hypernym that does not explicitly correspond to any synset. In fact, we are saying that we are looking for a synset hypernym, for example, for N = 5. Let's suggest that we haven't found it. We go down to N = 4, N = 3, N = 2, N = 1. Eventually, we can come to the point, where our output is littered with a few words (unigrams), which are equally "suitable" for the role of hypernym. In the worst case, it leads to a drop in precision. However, if the correct hypernym can be extracted from the definition, it would be found among the results. This problem was solved because the used method was complementing the others in such a way, that the method's accuracy itself was not critical. That's why it became possible to avoid its disadvantages.

We distinguish several types of errors.

- Words, that were not presented in the dictionary.
- Too general concept as a result. It happened because of errors, appearing during matching synset with taxonomy. If definition contains some right synset 'A B', but has it in form '<target> is A with ... B', it wouldn't match with the whole synset. More evidently that only A or B would take place in the answer. It can be right in some cases, especially if B is property of A, and 'A B' is 'A', but it wouldn't be right in the full sense.
- Skip-grams. The phrase corresponding to the correct hypernym is not contiguous. I.e. N-gram was broken by additional words.

- Rejected examples. If the result has more than 4 alternatives or more than 50% words from definitions, it is more likely to be wrong. Such results were considered inconsistent and have been ignored.
- Deformed correct synset. Method hadn't recognized right synset because of it's form. This type of error emerged because of system errors. The method retrieved some hypothesis, but no one of them was right. It was occurred due to the sliding window, that stopped before the right short answer and returned wrong, but long synset.
- Extensional definitions. Such definitions formulates its meaning by specifying term extension. We found some ostensive definitions, that had only one example or some quotes. For example, we have the definition *Полевод — тот, кто занимается полеводством, возделывает зерновые, технические, кормовые и бахчевые культуры* (*Field crop is the one who practices agriculture*). The DP method can not retrieve any appropriate synset from this definition.

Table 8: Type of errors for DP method

Type of error	%	word	definition	result before matching
Words were not presented in the dictionary	43%			
Too general concept	14%	криль	«зоол. промысловое название планктонных морских рачков»	'НАЗВАНИЕ'
Skip-gram	10%	мастиф	«древняя английская порода догообразных сторожевых собак»	'ПОРОДА'
Rejected examples	8%	сонник	«устар. книга, содержащая толкование снов Пример употребления: То вдруг велит науки прекратить, а молодых людей исключительно с одними сонниками знакомить, а потом, смотришь, сонники в печку полетели, а науки опять в чести сделались. М. Е. Салтыков-Щедрин, „В среде умеренности и аккуратности"»	'КНИГА', 'ТО', 'НАУКИ', 'ПРЕКРАТИТЬ', 'ИСКЛЮЧИТЕЛЬНО', 'ЗНАКОМИТЬ', 'ПОТОМ'
Deformed correct synset	16%	книгочей	«устар. любитель книг, чтения, знаний; книжник Пример употребления: Буфетная прислуга стала смотреть на меня исподлобья, мне говорили: — Эй ты, книгочей! Ты за что деньги получаешь? Максим Горький, „В людях"»	'МАКСИМ', 'ГОРЬКИЙ'
Extensional definitions Ostensive definition	6%	догматик гирька	«Значение ... 2. тот, чьё мышление отличается догматизмом» «Пример употребления: Раз только Соломин рассердился не на шутку и так ударил своим могучим кулаком по столу, что всё на нём подпрыгнуло, не исключая пудовой гирьки, приютившейся возле чернильницы. И. С. Тургенев, „Новь"»	'ТОТ' 'РАЗ', 'ТОЛЬКО', 'НЕ', 'НА', 'И', 'ПО', 'ЧТО', 'ВСЁ', 'НА', 'НЕ', 'ИСКЛЮЧАЯ'

References

1. *Amrami, A., Goldberg, Y.:* Towards better substitution-based word sense induction. CoRR. abs/1905.12598, (2019).
2. *Arefyev, N. et al.:* Evaluating three corpus-based semantic similarity systems for russian. In: Proceedings of the 21st international conference on computational linguistics and intellectual technologies (dialogue'2015). (2015).
3. *Conneau, A. et al.:* Unsupervised cross-lingual representation learning at scale. arXiv preprint arXiv:1911.02116. (2019).
4. *Devlin, J. et al.:* BERT: Pre-training of deep bidirectional transformers for language understanding. In: 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). pp. 4171–4186 Association for Computational Linguistics, Minneapolis, Minnesota (2019).
5. *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).
6. *Levy, O., Goldberg, Y.:* Neural word embedding as implicit matrix factorization. In: Ghahramani, Z. et al. (eds.) NIPS. pp. 2177–2185 (2014).
7. *Loukachevitch, N. V. et al.:* Creating russian wordnet by conversion, (2016).
8. *Lyons, J.:* Semantics. Cambridge University Press (1977).
9. *Mikolov, T. et al.:* Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR. 2013, (2013).
10. *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).
11. *Zesch, T. et al.:* Using wiktionary for computing semantic relatedness. Proceedings of the National Conference on Artificial Intelligence. 2, 861–866 (2008).