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

Moscow, June 17–20, 2020

HYBERT TEAM AT DIALOGUE EVALUATION 2020: TRANSFORMER FOR HYPERNYM EXTRACTION

Trofimova M. V. (mary.vikhreva@gmail.com),
Arkhipov M. U. (arkhipovmu@gmail.com)

Neural Networks and Deep Learning Lab; Moscow Institute of Physics and Technology, Moscow, Russia

The report describes the system developed by the HyBert team for the taxonomy enrichment task at Dialog Evaluation 2020 [12]. In this work we investigate the ability of large pre-trained language models to discover hyponym-hypernym relations. We probed state-of-the-art Transformers on the hypernymy task to evaluate implicit hierarchical knowledge captured during self-supervised training. To do so we use a simple distance-based classifier on the representations produced by the Transformer. Furthermore, we examine the performance of supervised approaches with a wide range of different training and embedding strategies. We show that while being a high capacity model, the Transformer is surprisingly hard to train to resolve hypernymy.

Key words: hypernym extraction, taxonomy enrichment, dialogue evaluation, Russian

DOI: 10.28995/2075-7182-2020-19-1170-1179

КОМАНДА HYBERT НА DIALAGUE EVALUATION 2020: ТРАНСФОРМЕР ДЛЯ ЗАДАЧИ ИЗВЛЕЧЕНИЯ ГИПЕРОНИМОВ

Трофимова М. В. (mary.vikhreva@gmail.com),
Архипов М. Ю. (arkhipovmu@gmail.com)

Лаборатория нейронных систем и глубокого обучения; Московский физико-технический институт, (национальный исследовательский университет), Москва, Россия

1. Introduction

The task of hypernym extraction is an important block for building and enriching semantic taxonomies such as WordNet [4]. The taxonomies can be further used to improve question answering [23] or query expansion [6]. But it is a very knowledge intensive and time consuming work to manually keep the taxonomies up-to-date or construct them for new languages and domains. Thus there is a strong need for automatic methods of hypernym extraction.

The recent language models like ELMo [13], GPT-2 [15] and BERT [3] have become widely used due to their ability to provide high quality contextualized word embeddings capturing a wide range of linguistic fenomena. The architectures are pretrained using only unlabeled text data, but, nevertheless, have been successfully applied to a variety of natural language understanding tasks and achieved state-of-the-art performance.

Our work is addressing the problem of hypernym extraction using only unstructured texts. We investigate the ability of the large pretrained language models to discover hyponym-hypernym relations. Specifically, we choose BERT from a wide range of available language models used as contextualized encoders. We choose it over GPT-2, because its architecture fits better for the non-autoregressive setting. We choose it over ELMo, because ELMo, which uses a bidirectional LSTM, simply concatenates the left-to-right and right-to-left information, while BERT's representation takes advantage of both left and right contexts simultaneously.

We evaluate our approach on a morphologically rich Russian language, on the task of enriching ruWordNet [11], semantic taxonomy for the Russian language, with hypernyms that are out-of-word due to introduction of neologisms. The implementation of our approach and pretrained models (`Torch`[1]) are available on github[2].

2. Dataset and Task

Taxonomies are tree structures which organize terms into a semantic hierarchy. Taxonomic relations (or hypernyms) are "is-a" relations: cat is-a animal, banana is-a fruit, Microsoft is-a company, etc. The goal of the Taxonomy Enrichment track of Dialogue Evaluation 2020 is to extend an existing taxonomy of ruWordnet [11] with relations of previously unseen words.

During evaluation phase participants are expected to predict a ranked list of 10 possible candidates for each new word in the test set. Particularly, candidates are supposed to be synsets of word senses, not just words. It is due to the ruWordNet initial structure: it contains synsets of word senses and relations between synsets, thus for each new word there is a need to predict its hypernym synset in the taxonomy. Evaluation is performed using the Mean Average Precision (MAP) and Mean Reciprocal Rank (MRR) scores.

Training data consists of (word, hypernym synset) pairs. For example, there are three hypernym synsets associated with the word 'МАРМЕЛАД' in the training set (see

[1] https://github.com/pytorch/pytorch

[2] http://github.com/vikmary/hyperhypo

Table 1). There is one first-order hypernym synset with the name ‘КОНДИТЕРСКОЕ ИЗДЕЛИЕ’, and also there are two hypernym synsets for the hypernym synset (which are hypernyms of second-order to the initial word ‘МАРМЕЛАД’) ‘ПРОДУКТЫ ПИТАНИЯ‘ and ‘СЛАДКОЕ КУШАНЬЕ’. The “Senses” column contains phrases representing a synset. All three synsets are considered true prediction for the word ’МАРМЕЛАД’.

Table 1: Hypernym synsets for the word ‘МАРМЕЛАД’

Synset name	Description	Senses
КОНДИТЕРСКОЕ ИЗДЕЛИЕ	—	ИЗДЕЛИЕ КОНДИТЕРСКОЙ, КОНДИТЕРКА, КОНДИТЕРСКОЕ ИЗДЕЛИЕ
ПРОДУКТЫ ПИТАНИЯ	продовольственные продукты, продовольствие	ПРОДУКТЫ, ПИЩЕВАЯ ПРОДУКЦИЯ, ПИЩЕВЫЕ ПРОДУКТЫ, ПРОДТОВАР, ПРОДУКТЫ ПИТАНИЯ
СЛАДКОЕ КУШАНЬЕ	—	СЛАДОСТИ, СЛАДКОЕ БЛЮДО, СЛАДКОЕ КУШАНЬЕ, СЛАСТИ

Compared to previous campaign for taxonomy enrichment SemEval-2016 task 14 [8], the participants of Dialogue Evaluation 2020 [12] are not given the definitions of words but only new unseen words in context. The organizers additionally provide news text corpus, parsed Wikipedia corpus and hypernym database from Russian Distributional Thesaurus.

3. System Description

3.1. BERT

BERT is a multi-layer bidirectional encoder based on the original Transformer architecture by [21]. The encoder is composed of a stack of 12 identical layers. Each layer has two sub-layers: a multi-head self-attention mechanism, and a simple, fully connected feed-forward network. A residual connection is also applied to each of the two sub-layers, followed by layer normalization. Input embeddings are a sum of three components: positional embeddings, which are cosine functions of different frequencies, trained token embeddings and segment identifier embeddings, which are learned for two possible values of 0 and 1. The later segment identifier is needed to separate sentences of different nature (sentence from the first source will have identifier equal to 0, and the second will have 1). A prominent feature of Transformer is that it takes as input tokens produced by WordPiece tokenizer (which are often subparts of actual words).

BERT is using masked language modeling objective in combination with next sentence prediction objective to train its weights including token embeddings.

We are exploring a BERT model called RuBERT[3] [10], which is a multilingual BERT with reassembled vocabulary of input tokens for the Russian language and with all layers fine-tuned on the Russian part of Wikipedia and news texts.

[3] https://github.com/deepmipt/DeepPavlov

3.2. Unsupervised Hypernym Extraction with BERT

3.2.1. Task

Unsupervised approach to hypernym extraction implies that for a given input word w having set of its contexts in a corpora $C_w = \{c_{w,1}, c_{w,2}, \dots c_{w,n}\}$ the model should predict list of its hypernym synsets of first and second order $HH = \{s_1, s_2, \dots s_m\}$ without training on labeled data.

3.2.2. Architecture

We do that by leveraging rich output BERT representations of texts containing input word. By taking contexts with the input word and embedding it with BERT, we then find the closest (in the BERT representations space) words from the target taxonomy that are embedded using the same BERT model. We then take hypernym synsets of the predicted words from the existing taxonomy relation graph, and output the the synsets as our predictions.

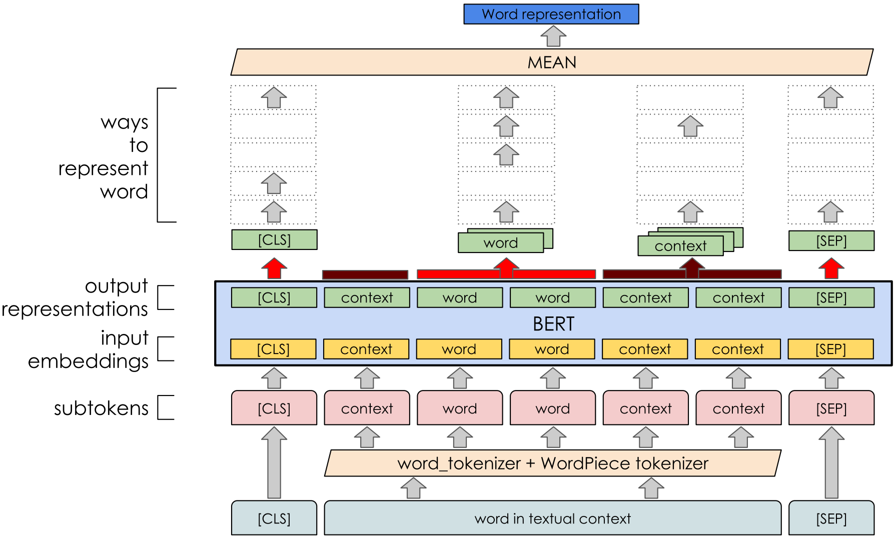

Figure 1: Constructing representation for input word using its context. Context with the word inside along with special tokens is split to WordPiece subtokens and fed to BERT, then output BERT representations are aggregated into a single vector which is the final representation of the input word. The figure depicts several alternative ways to aggregate output BERT representations: 1) average subtokens for the whole input, 2) take `[CLS]`'s output representation; 3) average subtokens corresponding to the input word; 4) average subtokens corresponding to the input word and the context; 5) average subtokens corresponding to the input word and special tokens.

Explicitly, we model (input word, word from taxonomy) pair in the following way: we encode input word as an average of its contexts $C_w = \{c_{w,1}, c_{w,2}, \dots c_{w,n}\}$ representations and encode word from the target taxonomy (the one to be enriched) w_t as:

$$y_w = \frac{\sum_i^n red_1(T(c_{w,i}))}{n}$$

$$y_{w_t} = red_2(T(w_t))$$

where T is a Transformer, $red_i(.)$ is a function that reduces sequence of output representations of Transformer to one vector. We experiment with various reduction functions for the input words $red_1(.)$ (see Experiments section). And we choose reduction function for the taxonomy words $red_2(.)$ to be the average of all Transformer output representations

$$red_2(v_{sbt_1}, v_{sbt_2}, \ldots v_{sbt_k}) = \frac{\sum v_{sbt_i}}{k}$$

where v_{sbt_i}—Transformer's output representation for i-th input subtoken.

3.2.3. Context and input word modeling

To represent input word and one of its contexts we construct a sample with the following format:

[CLS] context$_{left}$ word context$_{right}$ [SEP],

where context$_{left}$ are subtokens corresponding to context on the left side of the word, and context$_{right}$ contains subtokens corresponding to context on the right side of the word. word refers to subtokens of the input word, while [CLS] and [SEP] are special tokens.

3.2.4. Taxonomy word modeling

To represent a word from the taxonomy we construct a sample as:

[CLS] word [SEP],

where word refers to subtokens of the taxonomy word, while [CLS] and [SEP] are special tokens.

3.2.5. Scoring

We calculate scores for candidates from the taxonomy on two levels: word-level and synset-level.

To calculate score for a taxonomy word w_t we use dot-product:

$$s(w, w_t) = y_w^T y_{w_t}$$

To calculate score for a candidate synset s_t, which is a set of taxonomy words $s_t = \{w_{t,s_1}, w_{t,s_2}, \ldots w_{t,s_k}\}$, we first represent synset as an average of its word representations and then apply cosine distance:

$$y_{s_t} = \frac{\sum_i^k y_{w_{t,s_i}}}{k}, \ s(w, s_t) = \frac{y_w^T y_{s_t}}{\| y_w \| \| y_{s_t} \|}$$

3.2.6. Inference

Our approach is optimized to be infered super fast. It is achieved by beforehand calculation and caching of representations for all taxonomy synsets. If done so, we can do prediction by taking the closest taxonomy synsets in terms of cosine distance.

To enhance our predictions we take hypernyms of predicted synsets as final predictions. The enhancement is achieved by the fact that the Dialogue 2020 challenge's metric handles second-order hypernyms (hypernyms of hypernyms of input words) as true predictions along with first-order hypernyms (hypernyms of input words).

3.3. Supervised Hypernym Extraction with BERT

3.3.1. Task

Unsupervised approach to hypernym extraction implies that for a given input word w having set of its contexts in a corpora $C_w = \{c_{w,1}, c_{w,2}, \ldots c_{w,n}\}$ the model should predict list of its hypernym synsets of first and second order $HH = \{s_1, s_2, \ldots s_m\}$ using dataset of known (hyponym, hypernym synset) pairs.

3.3.2. Architecture

To use BERT for the hypernym extraction task in the supervised fashion:

1. we can take word-level BERT output representations as word embeddings or BERT output representation for `[CLS]` token as sentence embedding and train classifier on top;
2. alternatively, we can fine-tune the whole BERT on the downstream task.

The later approach is proved to be a better option for the BERT language model as it is reported in [14].

Thus, we encode input words and taxonomy words in the same way as described in the previous section, and fine-tune the whole BERT to output the representation for its hypernym words from the taxonomy.

3.3.3. Optimization

The network is trained using softmax loss to maximize score for true hypernym words with respect to all other words in the target taxonomy. For each pair of input word and its hypernym synsets in the training set $(w, HH(w)) = (w, \{s_1, s_2, \ldots s_m\})$ from the taxonomy T we compute loss as:

$$\mathcal{L}(w) = -\frac{1}{\sum_{j=1}^{m}|s_j|}\sum_{w_{t,i}\in\bigcup_{j=1}^{m}s_j} s(w, w_{t,i}) + \log\sum_{i=1}^{T}\exp s(w, w_{t,i}) \qquad (1)$$

However, we do not update pre-calculated representations of hypernym words from taxonomy due to high computational cost to calculate it at each update step.

4. Experiments

This section provides an overview of tested approaches in both supervised and unsupervised settings. We thoroughly investigated a wide range of methods to retrieve input word representations.

To gather mentions of a given word fast we construct an index of all available sources. To do so we perform tokenization with a simple regular expression which is identical to BERT Basic Tokenizer. Then we lemmatize each word with pymorphy2

[9] morphological analyzer. After this step, we build an index for all words in the WordNet and competition train/test sets. Using the index it is possible to extract all mentions of a given word in amortized constant time.

We use an embedding projection method from [20] paper as a baseline during our experiments.

We investigated different aggregation methods to discover best representations retrieval method $red_1(.)$. We tried averaging all embeddings presented in the mention sentence. However, a lot of tokens averaged might result in a quite noisy representation. So we also explored representations obtained from the `[CLS]` and only from subtokens of the word of interest. Further, we tried to incorporate context information by adding mean representation of left and right contexts. To get less noisy variant of the previous option we average only word and `[CLS]` and `[SEP]` tokens. We also tried to mask the word of interest to broaden the spectrum of possible predicted words with potential hypernyms and used the best setting from previously explored.

Going further, we performed supervised training with hypernym classification objective (1). For this task we also used the best approach of obtaining representation discovered in the unsupervised experiments. As mentioned in the Section System Description we used hypernyms of predicted synsets to enhance quality of the algorithm in all settings.

5. Results

This section presents the results of our experiments. All results are listed in the **Table 2**.

Table 2: Results of RuBert on public test set for nouns

Method	MAP	MRR
RuBERT baseline	0.1312	0.1449
Ustalov et al, 2017 [20]	0.2351	0.2690
fastText baseline	**0.4348**	**0.4729**
RuBERT with $red_1(.)$ = `MEAN(emb(word+context))`	0.1162	0.1311
RuBERT with $red_1(.)$ = `MEAN(emb([CLS]+word+context+[SEP]))`	0.1307	0.1507
RuBERT with $red_1(.)$ = `MEAN(emb(word))`	0.1899	0.2090
RuBERT with $red_1(.)$ = `emb([CLS])`	0.2635	0.2885
RuBERT with $red_1(.)$ = `MEAN(emb([CLS]+word+[SEP]))`	**0.2675**	**0.2926**
RuBERT masked with $red_1(.)$ = `MEAN(emb([CLS]+word+[SEP]))`	0.2239	0.2519
RuBERT trained with $red_1(.)$ = `MEAN(emb([CLS]+word+[SEP]))`	0.2718	0.2974
RuBERT trained masked with $red_1(.)$ = `MEAN(emb([CLS]+word+[SEP]))`	**0.2844**	**0.3155**

The first section of the table depicts baselines we used (Ustalove et al, 2017 [20]) along with baselines coming from organizers (RuBERT and fastText baseline).

The second section of the table provides a list of unsupervised experiments devoted to different reduction (aggregation) strategies for hyponym representation. The main aim of this section is to determine the best way of aggregation of unsupervised representations produced by RuBERT [10] architecture. As we can see, extraction of only word-related subtoken representations from the Transformer followed by averaging (RuBERT Baseline) is outperformed by word specific strategies with integration of the context with special tokens.

The third section of the **Table 2** is devoted to supervised approaches. We examined best aggregation strategies from previous section to explore how beneficial for the Transformer architecture could be fine-tuning on the target task. As can be seen from the table, supervised learning allows to boost the performance higher than recent approaches based on single word embedding projection [20]. However, still BERT model struggles to perform highly multi class classification for a wide range of hypernym candidates. One of possible causes of low results of BERT based approaches is the discrepancy between methods of representation generation: non-contextualized for hypernyms and contextualized for neologisms. Furthermore, representations of hypernyms are fixed during training, what possibly makes them hard to fit. This issue can be eliminated with trainable encoder and contextualized hypernyms.

6. Related Work

The automatic taxonomy extraction has begun with manual-written lexical-syntactic patterns [7]. The patterns were further learned to be extracted from corpora [19]. The problem of the pattern-based method is their sparsity, because there must be contexts where hyponym and hypernym co-occur. Recent work [17] is partly elevating the sparsity problem by using matrix decomposition propagating co-occurrence frequency values for unseen pairs of (hyponym, hypernym).

Methods based on distributional vectors also aim to overcome the sparsity issue. Typically, a binary classifier is trained for each candidate (hyponym, hypernym) pair to predict whether the pair has is-a relation [16].

There were also successful approaches of combining pattern-based and distributional-based method. One of the examples is HypeNet [18], where three embeddings are used as input to a binary classifier: candidate hyponym embedding, candidate hypernym embedding and a LSTM representation of a lexical syntactic path for the pair.

Classification-based methods are computationally expensive, because they need to be applied to all candidate (hyponym, hypernym) pairs during inference. The inference became much faster with projection learning. In the work of [5] the affine transformation is learned to map input word embedding to a hypernym embedding. The inference is then done simply by finding nearest neighbors in the space of hypernym words. The approach was further improved in [20] by training on hard negatives. The similar idea is used in [1], where the projection learning approach is extended with several hypernym output representations and combined with a pattern-based approach. The model has performed well on SemEval-2018 task.

7. Conclusion and Future Work

In this study, we evaluated the ability of a Transformer pre-trained on a large corpus of language-specific corpora to discover hyponym-hypernym relations. We significantly outperformed simple BERT baseline using different inputs to BERT and different aggregations of output representations. We showed that training BERT on the task of hypernym word embedding prediction with cross-entropy loss over all taxonomy words does not give much improvement. We see our future work as:

1. improving current solution with interaction based approach in which both hyponym context and candidate hypernym synset is passed to a single Transformer architecture like in siamese architectures [2]
2. building two-stage approach where the first stage is generation of a limited set of candidates and the second stage is reranking of candidates with a seamese-like architecture [22]
3. extention of hypernym representation with contextualized information
4. incorporating word descriptions, which are freely available in the Wiktionary[4], which allows to perform zero-shot hypernym detection similar to [22].

References

1. *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).
2. *Das, A. et al.:* Together we stand: Siamese networks for similar question retrieval. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers). pp. 378–387 (2016).
3. *Devlin, J. et al.:* BERT: pre-training of deep bidirectional transformers for language understanding. CoRR. abs/1810.04805, (2018).
4. *Fellbaum, C.:* WordNet. In: The encyclopedia of applied linguistics. American Cancer Society (2012).
5. *Fu, R. et al.:* Learning semantic hierarchies via word embeddings. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: Long papers). pp. 1199–1209, Association for Computational Linguistics, Baltimore, Maryland (2014).
6. *Gong, Z. et al.:* Web query expansion by wordnet. Presented at the August (2005).
7. *Hearst, M. A.:* Automatic acquisition of hyponyms from large text corpora. In: COLING 1992 volume 2: The 15th International Conference on Computational Linguistics. (1992).
8. *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).

[4] wiktionary.org

9. *Korobov, M.:* Morphological analyzer and generator for russian and ukrainian languages. In: International conference on analysis of images, social networks and texts. pp. 320–332, Springer (2015).
10. *Kuratov, Y., Arkhipov, M.:* Adaptation of deep bidirectional multilingual transformers for russian language. CoRR. abs/1905.07213, (2019).
11. *Loukachevitch, N. V. et al.:* Creating russian wordnet by conversion. In: Komp'juternaja lingvistika i intellektual'nye tehnologii. pp. 405–415, Rossiiskii Gosudarstvennyi Gumanitarnyi Universitet (2016).
12. *Nikishina, I. et al.:* RUSSE'2020: Findings of the First Taxonomy Enrichment Task for the Russian Language. In: Computational linguistics and intellectual technologies: Papers from the annual conference "Dialogue". (2020).
13. *Peters, M. E. et al.:* Deep contextualized word representations. CoRR. abs/1802.05365, (2018).
14. *Peters, M. E. et al.:* To tune or not to tune? Adapting pretrained representations to diverse tasks. CoRR. abs/1903.05987, (2019).
15. *Radford, A. et al.:* Language models are unsupervised multitask learners. (2019).
16. *Roller, S. et al.:* Inclusive yet selective: Supervised distributional hypernymy detection. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: Technical papers. pp. 1025–1036, Dublin City University; Association for Computational Linguistics, Dublin, Ireland (2014).
17. *Roller, S. et al.:* Hearst patterns revisited: Automatic hypernym detection from large text corpora. In: Proceedings of the 56th annual meeting of the association for computational linguistics (volume 2: Short papers). pp. 358–363, Association for Computational Linguistics, Melbourne, Australia (2018).
18. *Shwartz, V. et al.:* Improving hypernymy detection with an integrated path-based and distributional method. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 1: Long papers). pp. 2389–2398, Association for Computational Linguistics, Berlin, Germany (2016).
19. *Snow, R. et al.:* Learning syntactic patterns for automatic hypernym discovery. In: Saul, L. K. et al. (eds.) Advances in neural information processing systems 17. pp. 1297–1304, MIT Press (2005).
20. *Ustalov, D. et al.:* Negative sampling improves hypernymy extraction based on projection learning. CoRR. abs/1707.03903, (2017).
21. *Vaswani, A. et al.:* Attention is all you need. CoRR. abs/1706.03762, (2017).
22. *Wu, L. et al.:* Zero-shot entity linking with dense entity retrieval. arXiv preprint arXiv:1911.03814. (2019).
23. *Zhou, G. et al.:* Improving question retrieval in community question answering using world knowledge. In: IJCAI. pp. 2239–2245 (2013).