NLP_HZ at SemEval-2018 Task 9: a Nearest Neighbor Approach

Wei Qiu
Alibaba Group, China
qiuwei.cw@alibaba-inc.com

Mosha Chen
Alibaba Group,China
chenmosha.cms@alibaba-inc.com

Linlin Li
Alibaba Group, China
linyan.lll@alibaba-inc.com

Luo Si
Alibaba Group, China
luo.si@alibaba-inc.com

Abstract

Hypernym discovery aims to discover the hypernym word sets given a hyponym word and proper corpus. This paper proposes a simple but effective method for the discovery of hypernym sets based on word embedding, which can be used to measure the contextual similarities between words. Given a test hyponym word, we get its hypernym lists by computing the similarities between the hyponym word and words in the training data, and fill the test word's hypernym lists with the hypernym list in the training set of the nearest similarity distance to the test word. In SemEval 2018 task9, our results, achieve 1st on Spanish, 2nd on Italian, 6th on English in the metric of MAP.

1 Introduction

Hypernymy relationship plays a critical role in language understanding because it enables generalization, which lies at the core of human cognition (Yu et al. (2015)). It has been widely used in various NLP applications (Espinosa Anke et al. (2016)), from word sense disambiguation (Agirre et al. (2014)) to information retrieval (Varelas et al. (2005)) , question answering (Prager (2006)) and textual entailment (Glickman et al. (2005)). To date, the hypernymy relation also plays an important role in Knowledge Base Construction task.

In the past SemEval contest (SemEval-2015 task 17[1], SemEval-2016 task 13[2]), the "Hypernym Detection" task was treated as a classfication task, i.e., given a (hyponym, hypernym) pair, deciding whether the pair is a true hypernymic relation or not. This has led to criticisms regarding its oversimplification (Levy et al., 2015). In the SemEval 2018 Task 9 (Camacho-Collados et al., 2018), the task has shifted to "Hypernym Discovery" , i.e., given the search space of a domain's vocabulary and an input hyponym, discover its best (set of) candidate hypernyms.

In this paper, the content is organized as follows: Section 2 gives an introduction to the related work; Section 3 describes our methods for this task, including word embedding projection learning as the baseline and the nearest-neighbour-based method as the submission result; The experimental results are presented in Section 4. We conclude the paper with Section 5.

2 Related Work

The work of identifying hypernymy relationship can be categorized from different aspects according to the learning methods and the task formulization. The earlier work (Hearst (1992)) formalized the task as an unsupervised hypernym discovery task, i.e., none hyponym-hypernyms pairs (x, y) are given as the training data. Hearst (1992) handcrafted a set of lexico-syntactic paths that connect the joint occurrences of x and y which indicate hypernymy in a large corpus. Snow et al. (2004) trained a logistic regression classifier using all dependency paths which connect a small number of known hyponym-hypernym pairs. Paths that were assigned high weights by the classifier are used to extract unseen hypernym pairs from a new corpus. Variations of Snow et al. (2004) were later used in tasks such as taxonomy construction (Snow et al. (2006); Kozareva and Hovy (2010); Carlson et al. (2010)), analogy identification (Turney (2006)), and definition extraction (Borg et al. (2009); Navigli and Velardi (2010)).

A major limitation in relying on lexico-syntactic paths is the requirement of the cooccurence of the hypernym pairs. Distributional methods are developed to overcome this limitation. Lin (1998) developed symmetric similarity

[1] http://alt.qcri.org/semeval2015/task17/

[2] http://alt.qcri.org/semeval2016/task13/

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 909–913
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

measures to detect hypernym in an unsupervised manner. Weeds and Weir (2003); Kotlerman et al. (2010) employed directional measures based on the distributional inclusion hypothesis. More recent work (Santus et al. (2014); Rimell (2014)) introduces new measures, based on the distributional informativeness hypothesis. Yu et al. (2015); Tuan and Ng (2016); Nguyen et al. (2017) learn directly the word embeddings which are optimized for capturing the hypernymy relationship.

The supervised methods include Baroni and Lenci (2011); Roller et al. (2014); Weeds and Weir (2003). These methods were originally word-count-based, but can be easily adapted using word embeddings (Mikolov et al. (2013a); Pennington et al. (2014)). However, it was criticized that the supervised methods only learn prototypical hypernymy (Levy et al. (2015)).

3 Hyponym-hypernym Discovery method

3.1 Preprocessing

For the corpus and the train/gold/test data, we have two preprocessing steps: 1) Lowercase all the words; 2) Concatenate the phrases (hyponym or hypernym composed with more than one word) which occur in the training set or the test set with underline, i.e., "executive president" is replaced by "executive_president". It is quite useful for training word embedding models because we want to treat phrases as single words.

If there are multiple phrases in one sentence, we generate multiple sentences, one per phrase. For example, "executive president" and "vice executive president" both exist in the corpus sentence "Hoang Van Dung , vice executive president of the Vietnam Chamber of Commerce and Industry.". After preprocessing, two more sentences are generated and included in the training corpus for word embeddings:

- Hoang Van Dung , vice executive_president of the Vietnam Chamber of Commerce and Industry.
- Hoang Van Dung , vice_executive_president of the Vietnam Chamber of Commerce and Industry.

The size of the original corpus has increased after the preprocessing step, e.g., The English corpus has increased from ~18G to ~32G.

3.2 Word Embedding

We train our word embedding models using the Google word2vec (Mikolov et al. (2013a,b)) tool[3] on the preprocessed corpus. We employ the skip-gram model since the skip-gram model is shown to perform best in identifying semantic relations among words. The trained word embeddings are used in the projection learning and nearest-neighbour based method.

3.3 Method based on Projection Learning

The intuition of this method is to assume that there is a linear transformation in the embedding space which maps hyponyms to their correspondent hypernyms. We first learn a projection matrix from the training data, then apply the matrix to the test data. Our method is similar to that described in Fu et al. (2014), the main idea can be summarized as follows:

1. Give a word x and its hypernym y, assuming there exists a linear projection matrix Φ to meet $y = \Phi x$. We need to learn a approximate Φ using the following equation to minimize the MSE loss:

$$\Phi^* = \arg\min_{\Phi} \frac{1}{N} \sum_{(x,y)} \|\Phi x - y\|^2 \qquad (1)$$

2. Learn the piecewise linear projection by clustering the training data into different groups according to the vector offsets. The motivation for the clustering is two-fold: firstly, the hypernym-hyponym relation is diverse, e.g., offset from "carpenter" and "laborer" is distant from the one from "gold fish" to "fish"; Secondly, if a hyponym x has many hypernyms (or hierarchical hypernyms), we can't use a single transition matrix Φ to project x to different hypernym y. So a piecewise projection learning is needed in each individual group. Thus, the optimization goal can be formalized as follows:

$$\Phi_k^* = \arg\min_{\Phi_k} \frac{1}{N_k} \sum_{(x,y \in C_k)} \|\Phi_k x - y\|^2 \qquad (2)$$

Where N_k is the number of word pairs in the k^{th} cluster C_k.

[3] https://code.google.com/archive/p/word2vec/

3. Learn the threshold δ_k for each cluster, by assumming that positive (hyponmy-hypernmy) pairs can locate in radius δ while negative pairs can not:

$$d(\Phi_k x - y) = \|\Phi_k x - y\|^2 < \delta_k \quad (3)$$

Where d stands for the euclidean distance.

4. Once the piecewise projection and the threshold is learned, given a new hyponym x, all of the hypernym candidates ys from the vocabulary are paired with x. The pairs are assigned to the proper cluster by the vector offset (y-x). According to the threshold δ in that group, it can be decided whether (x, y) is a reasonable hyponym-hypernym pair.

3.4 Method Based on Nearest Neighbors

We noticed that the hypernyms are often very distant from the correspondent hyponyms in the embedding space. Meanwhile, hyponyms which are close to each other often share the same hypernyms. We propose a simple yet effective approach based on this observation.

Suppose the training set H consists of a number of hyponyms and their correspondent hypernyms

$$H : \{Hypo^k : Hyper_1^k ... Hyper_i^k\}$$

During the test time, for an unseen hyponym x, the top K nearest hyponyms in the training set , i.e., $Hypo_i$ are found, and their hypernyms are used as the output , i.e., the hypernyms of x. The found hypernyms are sorted according to the distance between x and $Hypo_i$. This can be formalized as follows:

$$\text{HypoN} = [\text{Hypo}_i].\text{sort\_by}(\text{distance}(\text{Hypo}_i, x))$$
$$\text{Hyper}(x) = [\text{Hyper}(w)|w \text{ in HypoN}]$$

where the $distance$ function measures the similarity between $Hypo_i$ and x, HypoN is the list of words from the training set sorted according to their distances to x. Consine similarity in the embedding space is used for the distance function in our setup. According to the requirements of Task 9, only the top 15 of $Hyper(x)$ are submitted for evaluation.

4 Evaluation

4.1 Experimental Setup

Word2vec is used to produce the word embeddings. The skip-gram model (**-cbow 0**) is used with the embedding dimension set to 300 (**-size 300**). The other options are by default. We use 10-fold cross validation to evaluate both methods on the provided training data. The results are shown in Table 1 [4]

4.2 Results Based on Projection Learning

For the projection learning method, we followed experimental settings described in Fu et al. (2014).The negative (hyponym, hypernym) pairs are randomly sampled from the vocabulary. The training set consists of the negative pairs and the positive pairs in 3:1 ratio.

By using the same evaluating metrics as PRF in the cited paper, our best F-value on the validation set is 0.68 (the paper result is 0.73) when the best cluster number is 2 and the threshold is (17.7, 17.3). We apply the learned projection matrices and thresholds on the validation data, extract out the candidate hypernyms from the given vocabulary and truncate the top 15 candidates by sorting them according to the $d(\Phi_k x, y)/\delta_k$ scores. The generated results are not very promising, see Table 1 for details.

This projection learning method performs not very well on task9, we think the most probable reason is that in Fu et al. (2014), the problem is formalized as a classification problem, in which the **(hyponym, hypernym)** pairs are given. However, our task is formalized as a hypernym discovery problem given only **hyponmys**. This task might be inherently much harder than the classification task; a second reason might be related to the relative small amount of training data, i.e., ~7500 training pairs in total.

4.3 Results Based on NN

The results are shown in Table 1 from row 2 to row 5. Table 2 shows the results evaluated on the test data. The performance evaluated using either cross validation or the test data is much worse than that of a typical hypernym prediction task reported by Weeds and Weir (2003). This illustrates that hypernym discovery is indeed a much harder task than the hypernym prediction task.

Although the method proposed by us is quite simple, our submissions are the 1st on Spanish, the 2nd on Italian, the 6th on English, ranked by the

[4] The PL based method is not evaluated on Italian or Spanish corpus due to its poor performance on English corpus. The result of PL method is not submitted for the task evaluation either.

System	Language	MAP	MRR	P@1	P@3	P@5	P@15
PL	English	2.8	7.6	7.5	3.3	2.6	2.0
NN	English	13.3	25.1	18.7	13.9	13.5	12.5
NN	Spanish	16.6	27.2	19.0	17.2	16.4	16.1
NN	Italian	19.3	32.4	25	19.8	18.6	18.6

Table 1: Cross validation results of the two methods on training set(%). PL stands for the projection-learning based system. NN stands for the nearest-neighbor based method.

Language	MAP	MRR	P@1	P@3	P@5	P@15
English	9.37	17.29	12	10.14	9.19	8.78
Spanish	20.04	28.27	21.4	20.95	20.39	19.38
Italian	11.37	19.19	13.1	12.08	11.23	10.9

Table 2: Results on the test data for our submissions(%).

metric of MAP. This proves the effectiveness of the method.

Compared with the results got by cross validation, the performance evaluated on the test data (Table 2) dropped significantly on English (MAP dropped by 4%) and Italian (MAP dropped by 8%), but increased by a margin on Spanish (MAP increased by 3.6%). We consider that it is due to the properties of provided data , i.e., the hypernyms in the test set are similar to those in the training set for Spanish, but dissimilar for English or Italian.

The performance drop for English and Italian exposes one of the main drawbacks of our method: the method can not discover the hypernyms that have never occurred in the training set. To overcome this shortcoming, using syntactic patterns to extract hyponym-hypernym with high confidence can be employed to enlarge the training set. We leave this to the future work.

5 Conclusion

In this paper we describe two methods we have tried out for the hypernym discovery task in SemEval 2018. We extended the method originally proposed for hypernym prediction by Fu et al. (2014) as a baseline system. However the performance of this method is poor. The nearest-neighbor-based method is relatively simple, yet quite effective. We analyzed the experimental results, reveal some shortcomings, and propose a potential extension to future improvement.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Comput. Linguist.*, 40(1):57–84.

Marco Baroni and Alessandro Lenci. 2011. How we BLESSed distributional semantic evaluation. *GEMS '11 Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics*, pages 1–10.

Claudia Borg, Mike Rosner, and Gordon Pace. 2009. Evolutionary algorithms for definition extraction. *Proceedings of the 1st Workshop on Definition Extraction*, pages 26–32.

Jose Camacho-Collados, Claudio Delli Bovi, Luis Espinosa-Anke, Sergio Oramas, Tommaso Pasini, Enrico Santus, Vered Shwartz, Roberto Navigli, and Horacio Saggion. 2018. SemEval-2018 Task 9: Hypernym Discovery. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, United States. Association for Computational Linguistics.

Andrew Carlson, Justin Betteridge, and Bryan Kisiel. 2010. Toward an Architecture for Never-Ending Language Learning. *In Proceedings of the Conference on Artificial Intelligence (AAAI) (2010)*, pages 1306–1313.

Luis Espinosa Anke, Jose Camacho-Collados, Claudio Delli Bovi, and Horacio Saggion. 2016. Supervised distributional hypernym discovery via domain adaptation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 424–435. Association for Computational Linguistics.

Ruiji Fu, Jiang Guo, Bing Qin, Wanxiang Che, Haifeng Wang, and Ting Liu. 2014. Learning semantic hierarchies via word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

1199–1209, Baltimore, Maryland. Association for Computational Linguistics.

Oren Glickman, Ido Dagan, and Moshe Koppel. 2005. A probabilistic classification approach for lexical textual entailment. In *AAAI*, pages 1050–1055. AAAI Press / The MIT Press.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. *Proceedings of the 14th conference on Computational linguistics -*, 2:539.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *Natural Language Engineering*, 16(4):359–389.

Zornitsa Kozareva and Eduard Hovy. 2010. A Semi-Supervised Method to Learn and Construct Taxonomies using the Web. *Proceedings of EMNLP, MIT, Massachusets, USA*, (October):1110–1118.

Omer Levy, Steffen Remus, Chris Biemann, and Ido Dagan. 2015. Do supervised distributional methods really learn lexical inference relations? *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976.

Dekang Lin. 1998. An Information-Theoretic Definition of Similarity. *Proceedings of ICML*, pages 296–304.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Distributed Representations of Words and Phrases and Their Compositionality. *In Advances in neural information processing systems*, pages 3111–3119.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546.

Roberto Navigli and Paola Velardi. 2010. Learning Word-Class Lattices for Definition and Hypernym Extraction. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010).*, (July):1318–1327.

Kim Anh Nguyen, Maximilian Köper, Sabine Schulte im Walde, and Ngoc Thang Vu. 2017. Hierarchical Embeddings for Hypernymy Detection and Directionality. pages 233–243.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

John Prager. 2006. Open-domain question: Answering. *Found. Trends Inf. Retr.*, 1(2):91–231.

Laura Rimell. 2014. Distributional Lexical Entailment by Topic Coherence. *In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden, April 26-30 2014*, pages 511–519.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet Selective: Supervised Distributional Hypernymy Detection. *In Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, August 23-29 2014*, pages 1025–1036.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte im Walde. 2014. Chasing Hypernyms in Vector Spaces with Entropy. *Proc. European Chapter of the Association for Computational Linguistics*, pages 38–42.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2004. Learning syntactic patterns for automatic hypernym discovery. *Advances in Neural Information Processing Systems 17*, 17:1297–1304.

Rion Snow, Daniel Jurafsky, and Andrew Y Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL - ACL '06*, pages 801–808.

Luu Anh Tuan and See Kiong Ng. 2016. Learning Term Embeddings for Taxonomic Relation Identification Using Dynamic Weighting Neural Network. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP-16)*, pages 403–413.

Peter D. Turney. 2006. Similarity of Semantic Relations. (March 2005):1–39.

Giannis Varelas, Epimenidis Voutsakis, Paraskevi Raftopoulou, Euripides G.M. Petrakis, and Evangelos E. Milios. 2005. Semantic similarity methods in wordnet and their application to information retrieval on the web. In *Proceedings of the 7th Annual ACM International Workshop on Web Information and Data Management*, WIDM '05, pages 10–16, New York, NY, USA. ACM.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 81–88.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. *IJCAI International Joint Conference on Artificial Intelligence*, 2015-January(Ijcai):1390–1397.