EXPR at SemEval-2018 Task 9: A Combined Approach for Hypernym Discovery

Ahmad Issa Alaa Aldine[1,3], **Mounira Harzallah**[2]
Berio Giuseppe[1], **Nicolas Béchet**[1], **Ahmad Faour**[3]
[1] IRISA - University Bretagne Sud, France
[2] LINA - University of Nantes, France, [3] Lebanese University, Lebanon
ahmad.issa-alaa-eddine@univ-ubs.fr
mounira.harzallah@univ-nantes.fr, giuseppe.berio@univ-ubs.fr
nicolas.bechet@irisa.fr, ahmad.faour@ul.edu.lb

Abstract

In this paper, we present our proposed system (EXPR) to participate in the hypernym discovery task of SemEval 2018. The task addresses the challenge of discovering hypernym relations from a text corpus. Our proposal is a combined approach of path-based technique and distributional technique. We use dependency parser on a corpus to extract candidate hypernyms and represent their dependency paths as a feature vector. The feature vector is concatenated with a feature vector obtained using Wikipedia pre-trained term embedding model. The concatenated feature vector fits a supervised machine learning method to learn a classifier model. This model is able to classify new candidate hypernyms as hypernym or not. Our system performs well to discover new hypernyms not defined in gold hypernyms.

1 Introduction

Hypernymy is an important lexical-semantic relation that is useful for many applications such as question answering, machine translation, information retrieval, and so on. In addition, hypernym relations are the backbone for building ontologies.

Various methods have been proposed to detect hypernym relation from text corpora. Most of these techniques are either path-based techniques or distributional techniques. In path-based methods, the detection of hypernym relations is based on the lexico-syntactic paths connecting a pair of terms in a corpus. Conversely, distributional methods are based on the distribution of term pair contexts. Most of these methods were unsupervised. Recently, focus shifted towards supervised methods.

This task inherits complexity and is far from being solved. The SemEval organizers address the same task but with a novel formulation (Camacho-Collados et al., 2018). They reformulate the task from hypernym detection into hypernym discovery. This novel formulation makes the task more realistic in terms of actual downstream application, while also enabling the benefits of information retrieval evaluation metrics. Hypernym detection focuses on deciding whether a hypernymic relation holds between a given pair of terms or not. Hypernym discovery focuses on discovering a set containing the best hypernyms for a given term from a given vocabulary search space. The task is divided into two subtasks: General-Purpose Hypernym Discovery and Domain-Specific Hypernym Discovery. The first consists of discovering hypernym in a general-purpose corpus, thus the SemEval organizers provide the participants with data for three languages: English, Italian, and Spanish. The second consists of discovering hypernym in a domain-specific corpus, thus they provide the participants with data for two specific domains: Medical and Music. The data contains a list of training terms along with gold hypernyms, a list of testing terms, and a vocabulary search space. The term is either a concept or an entity.

To tackle this task, we propose an approach that combines a path-based technique and distributional technique via concatenating two feature vectors: a feature vector constructed using dependency parser output and a feature vector obtained using term embeddings. Then, by using the concatenated vector we create a binary supervised classifier model based on support vector machine (SVM) algorithm. The model predicts if a term and its candidate hypernym are hypernym related or not.

2 Related Work

Most of the previous approaches for hypernymy detection are either path-based (patterns) or distributional based. Recently, some approaches are taking advantages of the combination of path-based and distributional techniques.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 919–923
New Orleans, Louisiana, June 5–6, 2018. ©2018 Association for Computational Linguistics

2.1 Path-Based

Path-based approaches are heuristic methods that predict hypernymy between a pair of terms if they match a particular pattern in a sentence of the corpus. These patterns are either manually identified (Hearst, 1992) or automatically extracted (Snow et al., 2005; Navigli and Velardi, 2010; Sheena et al., 2016). Approaches based on handcrafted patterns yield a good precision, but their recall is very low (Buitelaar et al., 2005). Approaches based on automatic learning of patterns achieve better performance by a small improvement in terms of precision and a considerable improvement in terms of recall, but the main limitation of these approaches is the sparsity of the feature space (Shwartz et al., 2016).

2.2 Distributional

Distributional approaches predict hypernym relations between terms based on their distributional representation, by either unsupervised or supervised models. The early unsupervised distributional models are based on symmetric measures (Lin, 1998). Later, asymmetric measures are introduced based on the Distributional Inclusion Hypothesis (DIH) (Weeds and Weir, 2003; Kotlerman et al., 2010). More recent, Santus et al. (2014); Rimell (2014) introduce new measures based on assumption that DIH is not correct for all cases. While, most of the supervised models rely on term embedding (Mikolov et al., 2013; Pennington et al., 2014) to represent the feature vector between the terms x and y. Various vector representations have been used such as concatenation $\vec{x} \oplus \vec{y}$ (Baroni et al., 2012) and difference $\vec{y} - \vec{x}$ (Roller et al., 2014; Weeds et al., 2014). More recent, Yu et al. (2015); Luu et al. (2016) suggested that models rely on term embedding are useful to indicate similarity between words, not to indicate hypernymy relations. Consequently, they learn their own term embedding models that are more relevant to indicate hypernym relations.

2.3 Combined Approaches

Combined approaches of distributional and lexico-syntactic paths are proposed based on the assumption that distributional approaches and path-based approaches have certain complementary properties. To our best knowledge, there are little works on integrating them (Mirkin et al., 2006; Kaji and Kitsuregawa, 2008). The recent work on integrating them is proposed by Shwartz et al. (2016). They use a long short-term memory (LSTM) network (Hochreiter and Schmidhuber, 1997) to encode dependency paths into a feature vector, then they concatenate the feature vector by the term embedding vectors of term x and term y.

3 System Description

As a preliminary step, we split each corpus into a training corpus and a testing corpus. Training corpus is a corpus of all sentences that contains training data terms (Concept/Entity), while testing corpus is a corpus of all sentences that contains testing data terms (Concept/Entity). Some sentences may contain training and testing data terms. These sentences will exist in both training and testing corpus.

3.1 Candidate Hypernyms

The first step in the system is to extract candidate hypernyms for the given training and testing data terms from a training corpus and a testing corpus respectively. We consider a term as a candidate hypernym if:

1. The term and its candidate occur in the same sentence.
2. The candidate exists in the vocabulary list.
3. The term and its candidate are noun phrases.
4. The term and its candidate are linked by short dependency path.

We consider a dependency path as short if it doesn't exceed two grammatical dependency relations. Using the short dependency path, we are capable representing paths similar to Hearst Patterns and other patterns. For example of short dependency paths, the dependency path between X and Y in the sentence S_1 "X such as Y" is {`nmod:such_as(`X `,` Y`)`} and in the sentence S_2 "X includes Y" is {`nsubj(includes , `X`), dobj(includes , `Y`)`}. We use Stanford dependency parser[1] (Marneffe et al., 2006) to extract dependency paths.

3.2 Feature Vector

The feature vector used to learn a model capable of predicting hypernym relations between a term

[1] https://stanfordnlp.github.io/CoreNLP/

and a candidate hypernym consists of the concatenation of two vectors: the first one is a vector extracted using a path-based technique while the second is extracted using a distributional technique.

The path-based vector consists of a set of features representing the short dependency path between a term y and its candidate hypernym x. The feature set is: $[Tag(x), GRel(x), HR, Freq, Tag(y), GRel(y)]$. $Tag(x)$ and $Tag(y)$ are the POS tag of x and y, $GRel(x)$ and $GRel(y)$ are the grammatical dependency relation of x and y, HR is the hypernym ratio of a dependency path and it is equal to the number of occurrences of a dependency path when indicating hypernm relation divided by the total occurrences of the same dependency path, and $Freq$ is the relative frequency of a dependency path and it is equal to the occurrence of a dependency path divided by the total occurrences of all dependency paths.

$$HR = \frac{hypernym\_DP\_occurrences}{DP\_occurrences}$$

$$Freq = \frac{DP\_occurrences}{Total\_DPs\_occurrences}$$

For a distributional based vector, We use pretrained 300 dimensional Word2Vec[2] term embeddings, trained on Wikipedia (Mikolov et al., 2013). We apply the difference between the embedding vector of term y and the embedding vector of term x ($\vec{y} - \vec{x}$)(Roller et al., 2014; Weeds et al., 2014). The term is either a single word or a multi-word expression.

3.3 Model Learning and Hypernym Discovery

In each training corpus, we extract a set of candidate hypernyms for each training term and label them if they are hypernym related or not using the gold hypernym data. Next, we represent each term and its candidate hypernym by a concatenated feature vector. These concatenated vectors are used for training the model. The classification method we used is SVM[3] with RBF kernel ($C = 1.0$, $gamma = 1/FeatureSize$). The training dataset was unbalanced, the ratio of hypernym instances w.r.t. not hypernym is less than 0.05. To represent the two categories (hypernym and not hypernym) in the training set, we improved this ratio to 0.2 by random elimination of not hypernym instances (20% hypernym instances and 80% not hypernym instances).

The classifier model is then used to discover hypernyms from a set of candidate hypernyms extracted from a testing corpus for each testing term by predicting if a term and its candidate hypernym are hypernym related or not. Each predicted hypernym is associated with a probability value. These values are used as ranking values to select the best fifteen hypernyms for each term (from higher to lower probability).

4 Results and Analysis

We submit our systems predictions for three corpora: English, Medical, and Music. The table 1 (a,b and c) below shows the result of our system and other supervised systems to discover hypernyms for Concept terms only. For the three corpora, our system performs better than STJU system, and it performs better than the MFH system on the English corpora. In addition, the result shows that our system performs well in discovering new hypernyms not defined in the gold hypernyms where it yields good False Positive values in the three corpora and we achieve the best False Positive value in Medical corpus (40) with a large difference to the second value (20) achieved by CRIM system.

Systems	MAP	MRR	P@1	P@3	P@5	P@15	False +
CRIM	**16.08**	**30.04**	**23.94**	**17.23**	**15.41**	**14.88**	20
MSCG	9.36	18.9	13.81	10.67	9.38	8.31	**28**
UMDuluth	8.13	18.93	15.33	8.83	7.53	7.07	20
NLP_HZ	7.17	13.13	8.99	7.69	7.11	6.71	24
Vanilla	6.99	16.05	12.3	7.69	6.55	6.18	
Begab	6.41	13.92	9.74	6.75	6.33	5.86	24
EXPR	*4.94*	*11.64*	*10.12*	*5.27*	*4.52*	*4.28*	*16*
MFH	4.73	12.48	11.92	4.84	4.13	3.93	
SJTU	3.29	5.68	0.28	3.45	3.57	3.54	0

(a) English Corpus.

Systems	MAP	MRR	P@1	P@3	P@5	P@15	False +
CRIM	**34.05**	**54.64**	**49.2**	**40.13**	**36.77**	**27.1**	20
MFH	28.93	35.8	32.6	34.27	34.2	21.39	
Begab	20.75	40.6	31.6	23.5	21.43	17.05	16
Vanilla	18.84	41.07	35.4	27.07	20.71	12.4	
EXPR	*13.77*	*40.76*	*38.2*	*17.17*	*12.76*	*9.34*	***40***
STJU	11.69	25.95	15.2	13.57	11.69	10.24	12

(b) Medical Corpus.

Systems	MAP	MRR	P@1	P@3	P@5	P@15	False +
CRIM	**43.38**	**63.79**	**52.79**	**47.16**	**43.87**	**40.14**	**24**
MFH	33.56	56.82	46.65	38.41	35.22	27.47	
Begab	23.52	39.26	24.02	23.23	22.66	23.13	16
Vanilla	11.53	35.78	31.28	13.59	10.28	8.46	
EXPR	*6.74*	*20.15*	*15.64*	*9.22*	*6.65*	*4.64*	*20*
SJTU	4.71	9.15	2.23	4.98	4.91	4.67	4

(c) Music Corpus.

Table 1: The evaluation results of our system and other supervised systems.

[2] https://radimrehurek.com/gensim/

[3] We use a machine learning python library scikit-learn (http://scikit-learn.org/stable/)

Our system result was beneath the expectation. By a short look into the output result files, we notice a lot of empty lines, meaning that our system was unable to discover any hypernym for a lot of terms and unexpectedly these terms correspond to all entity terms. In other words, our system lacks the ability to discover hypernyms for entity terms.

The table 2 (a,b and c) below shows the coverage of Wikipedia pre-trained term embedding model (TEM) and the coverage of candidate hypernym extraction (CHE) for the training and testing terms of the three corpora (English, Medical, and Music). The table shows that our system is unable to discover hypernyms for a considerable number of terms due to two main reasons. The first reason is that Wikipedia pre-trained term embedding model is limited in coverage, where many terms (Concepts/Entities) are not covered by the pre-trained embeddings, which leads to failure to discover hypernyms for these terms. For example, the term embedding (TEM) coverage of Medical Testing terms is 249 (50%), which means the system is unable to discover hypernyms for 251 (50%) terms not covered by the pre-trained term embedding. The second reason is that some conditions used to extract candidate hypernyms restrict the number of candidate hypernyms. For instance, the condition of the existence of a short dependency link between the term and its candidate causes the system to miss many candidate hypernyms if they are not linked by a short dependency path with the terms. In addition, the term and its candidate hypernym must occur as noun phrases in the sentence. This condition leads to failure to extract candidate hypernyms for some entity terms that can't be identified as noun phrases in the corpus such as "Up All Night", "Someday Came Suddenly", "Now What", etc. As shown in the table 2, the candidate hypernym extraction (CHE) coverage for English testing terms is 950 (63%), that means our system is unable to extract any candidate hypernym for 550 (37%) terms (398 entities and 152 concepts).

Furthermore, our system suffers from a major computational issue when applied to a large corpus. Parsing the corpus took to long and failed to complete before the submission deadline. Approximately, we processed 50% sentences of English corpus and 80% sentences of Music corpus, while we processed all sentences of Medical corpus. This explains why the performance of our system on Medical corpus is better than its performance on the two others corpora.

Terms	Training			Testing		
	Total	TEM	CHE	Total	TEM	CHE
Concept	979	824 (84%)	825 (84%)	1057	862 (81%)	905 (86%)
Entity	521	361 (69%)	49 (9%)	443	298 (67%)	45 (10%)
Total	1500	1185 (79%)	874 (58%)	1500	1160 (77%)	950 (63%)

(a) English Corpus.

Terms	Training			Testing		
	Total	TEM	CHE	Total	TEM	CHE
Concept	500	151 (30%)	414 (83%)	500	249 (50%)	427 (85%)
Entity	0			0		
Total	500	151 (30%)	414 (83%)	500	249 (50%)	427 (85%)

(b) Medical Corpus.

Terms	Training			Testing		
	Total	TEM	CHE	Total	TEM	CHE
Concept	387	227 (59%)	344 (89%)	358	228 (64%)	330 (92%)
Entity	113	57 (50%)	45 (40%)	142	82 (58%)	62 (44%)
Total	500	284 (57%)	389 (78%)	500	310 (62%)	392 (78%)

(c) Music Corpus.

Table 2: The coverage of wikipedia pre-trained term embedding model and candidate hypernym extraction.

5 Conclusion

In this paper, we presented our proposed system (EXPR) that is a combination of path-based technique and distributional technique to participate in Hypernym Discovery task of SemEval 2018. In this work, two feature vectors were extracted and concatenated: the first one is obtained using dependency parser on sentences and the second vector is obtained using pre-trained term embedding. A supervised classifier model based on SVM is built using training dataset composed of concatenated vectors. This model is used to discover hypernyms for new terms. The result was good but didnt fulfill our ambition due to several issues.

Our future work is to improve our approach for hypernym discovery by solving several issues. We believe that relying on term embedding model learned from the corpus provided in this task may be a good choice. In addition, we will work on the definition of a new dependency links not only those defined in this paper. Also, we will work to propose an unsupervised approach by using sequential pattern mining technique to automatically extract frequent sequential pattern between hyponym terms and their given hypernyms from the corpus.

References

Marco Baroni, Raffaella Bernardi, Ngoc-Quynh Do, and Chung chieh Shan. 2012. Entailment above the word level in distributional semantics. *In EACL*, pages 23–32.

Paul Buitelaar, Philipp Cimiano, and Bernardo Magnini. 2005. Ontology learning from text: An overview. *In Ontology Learning from Text: Methods, Applications and Evaluation*, pages 3–12.

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.

M. A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. *In Proceedings of the 14th International Conference on Computational Linguistics*, pages 539–545.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Comput.*, 9(8):1735–1780.

Nobuhiro Kaji and Masaru Kitsuregawa. 2008. Using hidden markov random fields to combine distributional and pattern-based word clustering. *In COLING*, pages 401–408.

Lili Kotlerman, Ido Dagan, Idan Szpektor, and Maayan Zhitomirsky-Geffet. 2010. Directional distributional similarity for lexical inference. *NLE*, pages 359–389.

Dekang Lin. 1998. An information-theoretic definition of similarity. *In ICML*, pages 296–304.

Anh Tuan Luu, Yi Tay, Siu Cheung Hui, and See-Kiong Ng. 2016. Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 403–413.

Marie-Catherine De Marneffe, Bill MacCartney, and Christopher D. Manning. 2006. Generating typed dependency parses from phrase structure parses. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, pages 449–454.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *In NIPS*, pages 3111–3119.

Shachar Mirkin, Ido Dagan, and Maayan Geffet. 2006. Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. *In COLING and ACL*, pages 579–586.

Roberto Navigli and Paola Velardi. 2010. Learning word-class lattices for definition and hypernym extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 1318–1327, Stroudsburg, PA, USA. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. *In EMNLP*, pages 1532–1543.

Laura Rimell. 2014. Distributional lexical entailment by topic coherence. *In EACL*, pages 511–519.

Stephen Roller, Katrin Erk, and Gemma Boleda. 2014. Inclusive yet selective: Supervised distributional hypernymy detection. *In COLING*, pages 1025–1036.

Enrico Santus, Alessandro Lenci, Qin Lu, and Sabine Schulte Im Walde. 2014. Chasing hypernyms in vector spaces with entropy. *In EACL*, pages 38–42.

N. Sheena, Smitha M. Jasmine, and Shelbi Joseph. 2016. Automatic extraction of hypernym and meronym relations in english sentences using dependency parser. *In Procedia Computer Science*, pages 539–546.

Vered Shwartz, Yoav Goldberg, and Ido Dagan. 2016. Improving hypernymy detection with an integrated path-based and distributional method. *CoRR*, abs/1603.06076.

Rion Snow, Daniel Jurafsky, and Andrew Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. *MIT Press*, pages 1297–1304.

Julie Weeds, Daoud Clarke, Jeremy Reffin, David Weir, and Bill Keller. 2014. Learning to distinguish hypernyms and co-hyponyms. *In COLING*, pages 2249–2259.

Julie Weeds and David Weir. 2003. A general framework for distributional similarity. *In EMLP*, pages 81–88.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2015. Learning term embeddings for hypernymy identification. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, pages 1390–1397. AAAI Press.