NTNU-2 at SemEval-2017 Task 10: Identifying Synonym and Hyponym Relations among Keyphrases in Scientific Documents

Biswanath Barik, Erwin Marsi
Department of Computer Science
Norwegian University of Science and Technology
{biswanath.barik,emarsi}@ntnu.no

Abstract

This paper presents our relation extraction system for subtask C of SemEval-2017 Task 10: ScienceIE. Assuming that the keyphrases are already annotated in the input data, our work explores a wide range of linguistic features, applies various feature selection techniques, optimizes the hyper parameters and class weights and experiments with different problem formulations (single classification model vs individual classifiers for each keyphrase type, single-step classifier vs pipeline classifier for hyponym relations). Performance of five popular classification algorithms are evaluated for each problem formulation along with feature selection. The best setting achieved an F_1 score of 71.0% for synonym and 30.0% for hyponym relation on the test data.

1 Problem Description

Task C of ScienceIE at SemEval-2017 (Augenstein et al., 2017) concerns identifying sentence level 'SYNONYM-OF' (or 'same-as') and 'HYPONYM-OF' ('is-a') relations among three types of *keyphrases*: PROCESS (PR), TASK (TA) and MATERIAL (MA) in scientific documents. The 'SYNONYM-OF' relation is symmetric, whereas the 'HYPONYM-OF' relation is directed. Hyponym relation prediction is thus associated with two ordered subtasks: (1) predicting relations between pairs of keyphrases; (2) predicting the direction of the relation. It is assumed that there are no relations between keyphrase of different types. Automatic identification of synonym/hyponym relations is useful for many NLP applications, e.g. knowledge base completion and ontology construction.

2 Challenges

The relation prediction task of ScienceIE is challenging and quite different from other semantic relation prediction task like SemEval-2010 Task 8 (Hendrickx et al., 2009). In SemEval-2010 Task 8, there are two marked nominals in a sentence and the task is to predict if any of nine semantic relations hold between the nominal pair. Although there are more relations than ScienceIE (9 vs 2), ScienceIE poses different challenges. Instead of single-word nominals, the keyphrases of ScienceIE are arbitrarily large text spans referring to larger syntactico-semantic units. The top part of Table 1 shows the percentage of keyphrases longer than 10 tokens in the training (10.89%), development (8.76%) and test (6.71%) data. The problem with such large text spans is to identify features which best represent the keyphrase and contribute most to the relation prediction task.

Another challenge of ScienceIE is the occurrence of multiple keyphrases in one sentence, producing a large number of possible relations among keyphrase pairs, i.e., $n(n-1)/2$ for n keyphrases. As most of these are negative instances, the positive and negative classes are imbalanced.

A third challenge is the potentially long distance between keyphrase pairs. The middle part of Table 1 shows that there are 49.2%, 57.68% and 43.77% keyphrase pairs in training, development and test sets respectively which are separated by more than 19 tokens. In addtion, a number of other keyphrases can occur in between a pair of related keyphrases, as shown in Table 1.

Finally,the number of synonym and hyponym relations in the training and development datasets is limited. The bottom part of Table 2 shows the frequencies of relations in training and development datasets (ignoring inter-sentence keyphrase relations).

Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017), pages 965–968,
Vancouver, Canada, August 3 - 4, 2017. ©2017 Association for Computational Linguistics

Table 1: Keyphrase related statistics on data sets

keyphrase length (ℓ)	**train**	**dev**	**test**
$\ell = 1$ (single word)	8.49	13.13	12.87
$2 \leq \ell \leq 5$	58.11	58.08	63.44
$6 \leq \ell \leq 10$	22.51	20.03	16.98
$\ell \geq 11$	10.89	8.76	6.71
inter-keyphrase distance (λ)	**train**	**dev**	**test**
$\lambda = 0$ (adjacent)	0.05	0.02	0.06
$1 \leq \lambda \leq 10$	20.60	16.17	22.24
$11 \leq \lambda \leq 20$	29.52	26.13	32.94
$\lambda \geq 20$	49.82	57.68	43.77
# intervening keyphrases (n)	**train**	**dev**	**test**
$n = 0$ (adjacent)	51.40	43.14	55.53
$n = 1$	23.84	23.95	25.13
$n = 2$	11.64	12.84	11.30
$n = 3$	5.64	7.32	4.57
$n \geq 4$	7.48	12.72	3.46

3 Approach

Inspired by the best systems at SemEval-2010 Task 8 (Rink and Harabagiu, 2010), we developed our relation extraction system in a supervised learning framework with the dependency structure of the input sentence as the major resource. The main intuition is that Bunescu and Mooney (2005) showed that the shortest path between two entities in a dependency graph contains most of the information for identifying the relation between them. In causal relation extraction (Barik et al., 2017), we have experienced that such intuition is effective. We tried two alternative approaches.

Approach-1: Individual vs Single Classifier As relations only occur between keyphrases of the same type, our first experiment evaluates the performance of separate synonym and hyponym classifiers for each keyphrase type, resulting in six classification problems. The description of System-1 provides more details on the classifiers.

The main challenge of developing individual classifiers for each task is the limited number of instances in the dataset. For example, there are only 11 relation instances between TASK (TA) keyphrases in the training data and only a single one in the dev data. Hence individual classifiers might not generalize well enough. Therefore, an alternative approach is to train one synonym classifier and one hyponym classifier for all keyphrase pairs, ignoring their types. This gives a higher number of positive training instances – 249 for synonym and 414 for hyponym – as shown in Table 2. This is the approach taken with System-2.

Table 2: Relation related statistics on data sets

Relation Type	**Dataset**	**PR**	**TA**	**MA**	Total
SYNONYM	train	150	11	88	249
SYNONYM	dev	23	1	21	45
HYPONYM	train	188	48	178	414
HYPONYM	dev	41	8	71	120

In both of these problem formulations, synonym is a binary classification problem, whereas the hyponym relation is considered as ternary classification (i.e., *forward* relation, *backward* relation and *no* relation).

Approach-2: Hyponym Relation-Direction Prediction Since the hyponym relation is directed, another option is to predict its direction separately. Whereas in Approach-1 hyponym relations and their direction were predicted simultaneously as a three class problem, in Approach-2 we have developed two systems – for relation prediction and direction prediction – and connect them in a pipeline. System-3 thus refers to a pipelined classification of hyponym relations.

4 Experiments

Preprocessing Input text is linguistically analyzed with the Stanford CoreNLP library (Manning et al., 2014), which includes sentence boundary detection, tokenization, lemmatization, part-of-speech (POS) tagging and dependency parsing.

Feature Extraction Features are extracted for every possible keyphrase pair within a sentence. The feature extraction process dependents heavily on contextual information and dependency structures, specifically, the shortest dependency path between two keyphrase heads and the dependency subtree connecting two keyphrases as described in (Liu et al., 2015). The major feature categories are:

- *context features*: bag-of-word – unigram & bigram, lemma, POS, word-POS combination
- *before & after context features*: bag-of-word – unigram & bigram, lemma, POS, word-POS combination in certain window sizes
- *dependency features*: dependency head & dependents of each keyphrase of the considered pair, head of the in-between context, dependency path between two entity heads, ordering of keyphrases in dependency path, dis-

Table 3: Candidate Classification Algorithms

Sl.	**Classification Algorithm**	**Parameters**
1	Support Vector Machines (SVM)	C, w, loss
2	Multinomial Naive Bayes (MNB)	Alpha
3	Decision Tree (DT)	split, w, max_feat
4	Random Forest (RF)	n_est., w, criterion
5	k-Nearest Neighbours (kNN)	N, weight
Sl.	**Feature Selection Method**	**Parameters**
1	χ^2-based feature selection (X2)	k
2	Tree-based feature selection (TR)	ExtraTreesClf
3	Recursive Feature Elim. (REF)	SVM

tance between two keyphrase heads in a dependency path

- *other features*: open bracket in the context, capitalization in keyphrase, length of keyphrase, number of lemma common to both keyphrase, number of intervening keyphrases
- *intervening keyphrase features*: the intervening keyphrase features like head of the keyphrase, its relation with context head, etc.
- *WordNet features* : synonym/hyponym relation between heads of two keyphrases, lexical cues for synonym/hyponym relation, e.g., 'such as', 'is a', 'including' etc.

Classifiers Used Instead of choosing any particular classification algorithm, we have evaluated five different classifiers with hyper-parameters and class weights tuned for different systems, as listed in the top half of Table 3.

Feature Selection Methods As shown Table 1, the keyhrase length (ℓ) and the in-between context length (λ) can be arbitrarily large. As a result, the feature extraction process generates a large number of features, many of which are unlikely to provide any useful information. Therefore we investigated three different feature selection techniques, as shown in the bottom half of Table 3. Among these feature selection techniques, χ^2-based feature selection (X2) gave the best result.

Parameter Optimization through CV The training instances were extracted from 350 training files, indexed by training file name, followed by preprocessing and feature extraction as described above. The class weights, parameters for five classifiers and k (the top-k feature for χ^2-based feature selection) were optimized for the three different experimental setups (System 1-3) descibed below using five fold cross validation with *grid search*, where training instances from the same training file are always in the same fold. Our implementation relied on classifiers, feature selection methods and CV grid search from Scikit-learn[1].

System-1 We ran CV experiments to optimize settings for the separate relation prediction tasks: synonym_process (SP), synonym_task (ST), synonym_material (SM), hyponym_process (HP), hyponym_task (HT) and hyponym_material (HM). For each task, we optimized the hyper-parameters of five classifiers as shown in Table 3. The performance of the best classifier was then evaluated on the development dataset. For the hyponym relation, we optimized on the micro-average score over the forward and backward relation.

System-2 System-2 consists of a combination of one synonym classifier and one hyponym classifier.

System-3 Hyponym relations and their directions were predicted by separate classifiers connected in a pipeline. Parameters were therefore optimized for relation and direction prediction separately. The synonym predictions of System-3 result from the combination of the synonym classifier of 1-4 and 2 where any keyphrase pair predicted by either classifier 1-4 or classifier 2 is considered as synonym.

5 Results

Table 4 shows the result of System 1-3 on development data, while Table 5 shows performance on test data. According to Table 4, the combined performance of individual classifiers (of System-1) for synonym (SM-SP-ST) and hyponym (HM-HP-HT) is 77% and 29%, which is slightly lower then the corresponding performance of system-2. This is consistent with performance on the test data.On the other-hand, the pipeline of System-3 shows a lower score than System-1 and System-2 for the hyponym relation.

5.1 Error Analysis

We have analyzed the mistakes produced by System 1-3 and found the following frequent error categories:

- *synonyms* - The synonyms with pattern KEYPHRASE1 (KEYPHRASE2 in abbrevi-

[1] http://scikit-learn.org/stable/

Table 4: Result of individual classifiers where hyponym relations are considered as three class problem with micro average of positive classes

Sys	Relation	Clf	Pr	Re	F_1
1-1	SM	SVM	0.93	0.62	0.74
1-2	SP	DT	0.78	0.78	0.78
1-3	ST	DT	1.00	1.00	1.00
1-4	SM-SP-ST	SVM-DT-DT	0.84	0.71	**0.77**
1-5	HM	RF	0.39	0.21	0.27
1-6	HP	SVM	0.51	0.27	0.35
1-7	HT	SVM	0.04	0.10	0.06
1-8	HM-HP-HT	RF-SVM-SVM	0.40	0.23	**0.29**
2	Syno	SVM	0.80	0.77	**0.78**
2	Hypo	DT	0.37	0.28	**0.32**
3	Syno 1-4+2	SVM	0.84	0.79	**0.81**
3	Rel	SVM	0.64	0.35	0.45
3	Dir	SVM	0.73	0.72	0.72
3	Rel → Dir	SVM-SVM	0.36	0.21	**0.26**

Table 5: Result of synonym and hyponym relation of System 1-3 on test data

System	Hyponym			Synonym		
	Pr	Re	F_1	Pr	Re	F_1
1	0.34	0.24	0.28	0.71	0.62	0.66
2	0.35	0.26	**0.30**	0.82	0.57	0.67
3	0.31	0.18	0.23	0.78	0.65	**0.71**

ation) like 'density of states (DOS)' are identified correctly. However, the opposite pattern like 'SRTM (Shuttle Radar Tropographical Mission)' are not well recognized.

- *hyponyms with conjunctions* - when a list of hyponyms is connected by conjunctions, often some hyponyms are missed.
- *hyponym to synonym* - In some cases hyponym patterns are quite similar to frequent synonym patterns and therefore misclassified. For example, in the sentence fragment, 'xR is the x-position of the receiving element (R)', the keyphrase 'R' is connected with 'receiving element' by a synonym relation, whereas the correct relation is hyponym.
- *synonym to hyponym* - In some cases a synonym relation is observed instead of a hyponym relation. For example, in 'constituent statistics (SB, SDSD, and LCS)', the keyphrases 'SDSD' and 'LCS' are correctly linked to the 'constituent statistics' by a hyponym relation, but 'SB' is incorrectly linked as a synonym.

6 Conclusion

We have described our system for predicting synonym and hyponym relations between keyphrases within a feature-based supervised learning framework. We have developed three systems for the synonym and hyponym prediction tasks. Experiments showed that with a relatively small dataset, training a single classifier for synonym and hyponym works slightly better than training separate classifiers for each keyphrase type. We also found that a pipeline of classifiers for relation and direction prediction of hyponym relations is not effective compared with predicting relation and direction simultaneously. As future work, we can investigate the performance of neural network-based relation classification approaches (specifically Convolution and Recurrent Neural Networks).

References

Isabelle Augenstein, Mrinal Kanti Das, Sebastian Riedel, Lakshmi Nair Vikraman, and Andrew McCallum. 2017. SemEval 2017 Task 10: ScienceIE - Extracting Keyphrases and Relations from Scientific Publications. In *Proceedings of the International Workshop on Semantic Evaluation*. Vancouver, Canada.

Biswanath Barik, Erwin Marsi, and Pinar Öztürk. 2017. Extracting Causal Relations among Complex Events in Natural Science Literature. In *Proceedings of the 22nd International Conference on Natural Language & Information Systems (NLDB)*. Liege, Belgium.

Razvan C Bunescu and Raymond J Mooney. 2005. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*. pages 724–731.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*. pages 94–99.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. *arXiv preprint arXiv:1507.04646* .

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*. pages 55–60.

Bryan Rink and Sanda Harabagiu. 2010. UTD: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*. pages 256–259.