Learning the Distinctive Pattern Space Features for Relation Extraction

Daojian Zeng[1], Yubo Chen[1], Kang Liu[1], Jun Zhao[1], and Xueqiang Lv[2]

[1] National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
{djzeng,yubo.chen,kliu,jzhao}@nlpr.ia.ac.cn
[2] Beijing Key Laboratory of Internet Culture and Digital Dissemination Research
Beijing Information Science & Technology University
lxq@bistu.edu.cn

Abstract. Recently, Distant Supervision (DS) is used to automatically generate training data for relation extraction. As the vast redundancy of information on the web, multiple sentences corresponding to a fact may be achieved. In this paper, we propose pattern space features to leverage data redundancy. Each dimension of pattern space feature vector corresponds to a basis pattern and the vector value is the similarity of entity pairs' patterns to basis patterns. To achieve distinctive basis patterns, a pattern selection procedure is adopted to filter out noisy patterns. In addition, since too specific patterns will increase the number of basis patterns, we propose a novel pattern extraction method that can avoid extracting too specific patterns while maintaining pattern distinctiveness. To demonstrate the effectiveness of the proposed features, we conduct the experiments on a real world data set with 6 different relation types. Experimental results demonstrate that pattern space features significantly outperform State-of-the-art.

1 Introduction

There are huge amount of unstructured texts on the web, which constitute a huge and rapidly growing information repository. However, these kinds of unstructured textual information is only human readable, which hinders people from developing more compelling applications. So, there is an urgent need to automatically convert unstructured information into structured data which can be understood by machines.

As one way to extract structural information, relation extraction is to predict semantic relations between entities from texts [3]. Generally, the task of relation extraction can be defined as follows: given two entities e_1 and e_2 in the corresponding text T, we aim to identify the relation between these two entities based on diverse lexical, syntactic and semantic information.

As the vast redundancy of information on the web (e.g., A facts may state several times in different ways within multiple sentences), a natural avenue for our research is to infer semantic relations by considering the multiple sentences that contain the entity pairs. Taking the entity pair ⟨*Fanboys*, *Kyle Newman*⟩ in the following two sentences as an example:

1. *He is also known for his role in [Kyle Newman] 's film [Fanboys].*
2. *The award was presented by [Fanboys] director [Kyle Newman].*

The first sentence provides evidence not only for *directed_by* but also *written_by* and *produced_by*. Similarly, we cannot infer the relation *directed_by* only from the second sentence (e.g. *"[Google privacy] director [Alma Whitten] is stepping down after a tough three-year tenure"*). However, we can deduce the relation between *Fanboys* and *Kyle Newman* convincingly in combination with these two sentences.

As far as I am aware, there are two published studies leveraging the redundancy of information for relation extraction [11, 4]. Mintz et al. [11] inferred the relations by simply concatenating the multiple sentences that contain the entity pairs via a supervised machine learning paradigm. However, this method resulted in the poor performance due to the high dimensionality of the extracted lexical features. For instance, the dimension of feature vector exceed one million in MultiR[1]. To solve the high dimensionality of the feature vector, Bollegala et al. [4] proposed relation dual representation and entity pairs are represented as the distribution over patterns. However, this method has two main shortcomings. First, there may be noisy patterns, which will have a serious impact on the final features. Second, this feature representation may cause the *null pattern* problem when using a supervised paradigm. For example, "X work as Y" is one of the patterns constituting the pattern space while "X serve as Y" is not in the space.

In this paper, we proposed pattern space features to remedy the two defects mentioned above. Pattern space features are represented as the distribution over selected basis patterns. To address the noisy patterns, we first extract lexical-syntactic patterns that connect to the given entity pairs from all of the matched sentences and entity pairs. Next, the patterns with low distinctiveness are filtered out based on Discriminative Category Matching (DCM) theory [8] and the preserved patterns constitute basis patterns. Finally, different from Bollegala et al. [4] proposed method, each vector value of pattern space features is the similarity of entity pairs' patterns to basis patterns calculated by a shortest dependency path kernel function rather than the occurrence counts. Thus, the *null pattern* problem is avoided.

In summary, the contribution of this paper can be concluded as follows.

- We propose pattern space features to identify relation between two entities. Pattern space features can leverage various kinds of features from multiple sentences.
- To extract distinctive basis patterns, we propose a novel pattern extraction approach and use DCM theory to filter out noisy and ambiguous patterns.
- To solve *null pattern* problem, we define a shortest dependency path kernel function to measure the similarity between two patterns.

[1] http://www.cs.washington.edu/ai/raphaelh/mr/

2 Related Work

Relation extraction is important topics in information extraction. Many approaches have been explored in relation extraction, including bootstrapping, unsupervised relation discover and supervised classification.

Bootstrapping methods for relation extraction are attractive because they first only need a very small number of seed instances or patterns [1, 5]. Then, new patterns are extracted for subsequent iterations until convergence. The quality of the extracted relations depends heavily upon the initial seeds. In general, the resulting patterns often suffer from semantic drift and low precision.

The second scheme is purely unsupervised relation extraction. The Distributional Hypothesis [9] theory indicates that words occur in the same context tend to have similar meanings. Accordingly, it is assumed that the entity pairs occur in similar context tend to have similar relations. In general, unsupervised relation extraction methods use contextual features to represent relation of entity pairs in large amounts of corpus, and then cluster these features to classify these entity pairs. Finally, the words between two entities are simplified to produce relation-strings. Hasegawa et al. [10] adopted a hierarchical clustering method to cluster the contexts of entity pairs and simply select the most frequent words in the contexts to represent the relation that holds between the entities. Rosenfeld et al. [14] and Bollegala et al. [4] proposed relation dual representation in which the entity pairs were represented as the distribution over pattern space. The entity pairs are clustered in the entity pair vs. pattern matrix to identify semantic relations. Unsupervised approach can use very large amounts of data and extract very large number of relation instances, but do not output canonicalized relations and need to do relation mapping for further usage.

In the supervised paradigm, relation extraction is considered as a classification problem and researchers have done much work on how to automatically derive feature to represent the relation between two entities. Generally, the methods can be categorized into two types: feature-based and kernel-based. In feature-based method, a diverse set of strategies have been exploited to convert the extraction clues in structures such as sequences, parse trees to feature vectors for use by classifiers [15]. Feature-based method suffers from the problem of selecting a suitable feature-set. Kernel methods provide a natural alternative to exploit rich representation of the input extraction clues like syntactic parse trees etc. Kernel methods allow the use of a large set of features without the need to extract them explicitly. So far various kernels have been proposed to solve relation extraction problem, such as convolution tree kernel [12], subsequence kernel [6] and dependency tree kernel [7]. The methods mentioned above, however, suffered from lacking of a large amount of labeled data for training. Mintz et al. [11] proposed DS to address this problem. The DS paradigm selects the sentences that match the facts in knowledge base as positive examples. DS algorithm sometimes exposes to wrong label problem and brings noisy labeled data. To address the shortcoming of DS, Riedel et al. [13] cast the relaxed DS assumption as multi-instance learning. Supervised paradigm has been demonstrated to be effective for relation detection and yields relatively high performance. In this

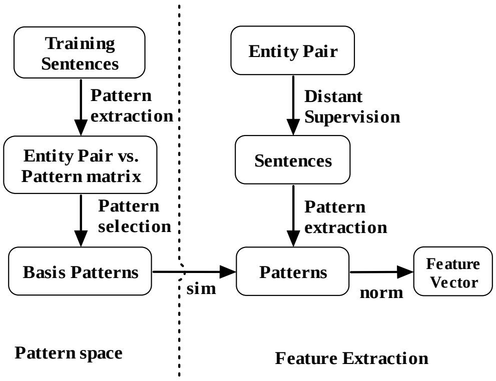


Fig. 1. The flow chart of pattern space feature extraction.

paper, we also employ supervised paradigm and DS is adopted to automatically generate training data. The wrong label problem in DS is ignored and beyond the scope of this paper. As discussed in section 1, it is very important to leverage information from multiple sentences. In this paper, we propose pattern space features to extract relation from multiple sentences based on relation duality and compare our feature with Mintz et al. [11] and Bollegala et al. [4] proposed feature in the experiment section.

3 Pattern Space Feature Extraction

3.1 Feature Generation Framework

In relation extraction task, our goal is to identify relation between two entities. The key problem is to pick up appropriate feature vector to represent relation of entity pairs, with a suitable measure of distance between vectors. The feature vector must be chosen in such a way that entity pairs have similar relation would be close to each other, and conversely, entity pairs involved in distinct relation would be far apart.

The flow chart of pattern space feature extraction is shown in Figure 1. It mainly includes two parts: *Pattern space* and *Feature extraction.* In the *Pattern space* part, our main goal is to acquire basis patterns. Firstly, we extract raw patterns from the training sentences. Furthermore, the entity pairs and entire patterns form a matrix in which each row represents an entity pairs and each column stands for a unique pattern based on relation duality. Finally, we use pattern selection method to filter out noisy pattern, and the basis patterns are achieved. The entire basis patterns span pattern space for generating feature. In the *Feature extraction* part, it usually provides us with an entity pair. Then, we can get its corresponding patterns through DS and pattern extraction. To extract pattern space features, we calculate the similarity among the corresponding patterns and basis patterns. The pattern space feature vector is created with each dimension corresponding to a basis pattern and the vector value is the similarity calculated above. Finally, the pattern space feature vector is further normalized by its length.

3.2 Pattern Space Features Explanation

Relational duality was formally put forward by Bollegala et al. in 2010, which means that a relation could be represented extensionally and intensionally. An extensional definition of a relation is to list the entity pairs containing such relation. On the other hand, a relation could be defined by specifying the patterns that express such relation. For example, consider the relation *directed_by* between a film and a person. The extensional definition the relation of *directed_by* enumerates the entity pairs hold this relation. (e.g. (*Bwana Devil, Arch Oboler*), (*Das Boot, Wolfgang Petersen*), etc.) Words occurring in the same context tend to have similar meanings through the Distributional Hypothesis theory [9]. In relation extraction, it is reasonable to assume that the entity pairs occurring in similar patterns tend to have similar relations. Accordingly, the intensional definition the relation of *directed_by* needs to specify the lexical or syntactic patterns belongs to this relation such as "X is a film directed by Y", "X is suspense-thriller film from director Y", etc.

As the dual representation of semantic relation mentioned above, we can represent the dual property as a matrix with entity pairs in the rows and patterns in the columns. The matrix cell value is the occurrence counts of entity pair in the pattern space. In previous works [4, 2], each row serves as a feature vector for an entity pair. Each dimension of the vector corresponds to a pattern and the vector value is the pattern occurrence counts of entity pair. Although entity pairs with similar relation would be close to each other through such representation, there are mainly two challenges. First, there may be noisy patterns in the patterns space and the sparsity in feature vector is severe. Second, the variation of patterns are more severe than word tokens, the space spanned by patterns from training samples is not a complete space. The patterns extracted from the test instance may not exist in pattern space and cause *null pattern* problem. The severity of the problem will depend upon numerous factors such as the amount of the train instances as well as the similarity between the train and test instances. With the amount the training instances increasing, the number of patterns will further increase. The severity of the *null pattern* problem will be less, but not completely non-existent. In addition, it will aggravate the sparsity in feature vector.

Hence, to address the *null pattern* problem mentioned above, we design a kernel function to compute the similarity between two patterns in the pattern space feature extraction framework. The matrix cell value is then replaced by the similarity and each row acts as the feature vector. Furthermore, we use DCM [8] theory to rank the patterns of each relation. Then, the patterns with low distinctiveness are filtered out (See section 3.4). The preserved patterns constitute basis patterns, which span the pattern space. An entity pair is represented as a vector and each dimension is the weight in the pattern space. This feature representation allows us to compute the similarity of two entity pairs by comparing the distribution over pattern space.

3.3 Pattern Extraction

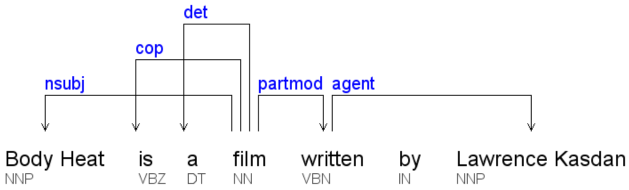


Fig. 2. The dependency tree of "[Body Heat] is a film written by [Lawrence Kasdan]".

The proposed feature generation algorithm regards the distribution of a particular entity pair over the space spanned by patterns as feature vector. Patterns play a key role in the feature extraction framework. Consequently, the pattern extraction module is crucial to the success of this approach. Too many patterns will cause the dimension of feature vector to be too large. It is not suitable to extract context pattern due to the wide variation of surface text. Fortunately, the syntactic structures of the sentence and the grammatical relations enable us to reduce the variation. Besides, syntactic features are indeed useful in relation extraction, especially when two entities are nearby in the dependency structure but distant in terms of words [11].

Following the idea in literature [7], we assume that the shortest dependency path tracing from entity 1 through the dependency tree to entity 2 gives a concrete syntactic structure expressing the relation between entity-pairs. For a given sentence, we first obtain its dependency tree using Stanford Parser[2]. Then, these dependencies form a directed graph, $\langle V,\ E\rangle$, where each word is a vertex in V, and E is the set of word-word dependencies, as shown in Figure 2. Then, we get the shortest connecting path between the given entity pair on the dependency graph to represent the relation. To avoid extracting too specific patterns, the lexical words in the shortest path are usually replaced with POS tags. However, the lexical form of verb is particularly important in relation extraction. For example, when the wildcard is replaced by different verb, the following pattern indicates distinct relationship(e.g. *written* vs. *written_by*, *directed* vs. *directed_by*, etc.).

$$\mathbf{X} \xleftarrow{nsubj} NN \xrightarrow{partmod} * \xrightarrow{agent} \mathbf{Y}$$

[2] http://nlp.stanford.edu/software/lex-parser.shtml

Hence, the lexical form of verb in the dependency path, which is distinctive to relations, is preserved in order to trade off data sparseness and pattern distinctiveness. According to the above analysis, the pattern extracted from Figure 2 is as follows:

$$\mathbf{X} \xleftarrow{nsubj} NN \xrightarrow{partmod} written \xrightarrow{agent} \mathbf{Y}$$

In order to compute the similarity between two patterns, we design a simple kernel function amounts to calculating the number of common features of two patterns, which is, to some extent, inspired by a shortest path dependency kernel proposed in paper [7]. If $p_x = x_1 x_2 \cdots x_m$ and $p_y = y_1 y_2 \cdots y_n$ are two patterns, where x_i and y_i denotes the set of tokens corresponding to position i of the patterns. The similarity of two patterns is computed as in Equation 1.

$$sim\,(p_x, p_y) = \begin{cases} 0, & m \neq n \\ \sum\limits_{i=1}^{n} t(x_i, y_i)/n, & m = n \end{cases} \tag{1}$$

Where $t(x_i, y_i) = 1$ if x_i equals y_i, else $t(x_i, y_i) = 0$. To better explain the definition of similarity function, let us consider the following sentence: "*In 2005, she appeared as flight attendant Claire Colburn alongside Orlando Bloom, in [Elizabethtown], a movie written and directed by [Cameron Crowe]* ". The corresponding pattern is:

$$\mathbf{X} \xrightarrow{appos} NN \xrightarrow{partmod} written \xrightarrow{agent} \mathbf{Y}$$

Then, this pattern is represented as a sequence set $p_1 = [x_1\ x_2\ x_3\ x_4\ x_5]$, where x_1 = "*appos*", x_2 = "*NN*", x_3 = "*partmod*", x_4 = "*directed*", and x_5 = "*agent*". Similarly, the pattern extracted from Figure 2 is represented as $p_2 = [y_1\ y_2\ y_3\ y_4\ y_5]$, where y_1 = "*nsubj*", y_2 = "*NN*", y_3 = "*partmod*", y_4 = "*written*", and y_5 = "*agent*". Based on the formula from Equation 1, the similarity between p_1 and p_2 is computed as $sim(p_1, p_2) = 4/5 = 0.8$.

3.4 Pattern Selection

The number of patterns in the pattern space is the final dimension of feature vector. Excessive number of patterns not only led to high computation cost but also feature sparsity. In the pattern space feature extraction paradigm mentioned above, the feature quality relies entirely on the distinctiveness of patterns. Filtering out noisy patterns is a matter of paramount importance.

In this paper, the DCM [8] theory is adopted to guide us select patterns and each relation type is analogy to a category in document collections. The approach proposed here for selecting patterns is based on two assumptions. The first is that patterns shared by majority of entity pairs in a relation are vital important for this relation. In other words, patterns that appear frequently within a relation category should be critical in term of classification, while patterns that shared only by few entity pairs are either less commonly used or provide implicit evidence of a relation. Following paper [2], to capture this property we define pattern significance, $Sig_{i,R}$, to weight pattern p_i within relation R.

$$Sig_{i,R} = \frac{\log_2 (N_{i,R} + 1)}{\log_2 (N_R + 1)} \tag{2}$$

Where $N_{i,R}$ is the number of pattern p_i in the cluster of relation R. N_R is the total number of patterns in Relation R.

The second assumption is that patterns shared by more than one relation category may be ambiguous and express different amounts of evidence to different relations. Patterns appear in more relations, more ambiguous it is. Similarly, we define the following Equation to capture the clarity of pattern p_i.

$$C_i = \begin{cases} \log_2 \frac{n * \max\limits_{j \in \{1 \cdot\cdot n\}} \{Sig_{i,R_j}\}}{\sum_{j=1}^{n} Sig_{i,R_j}} * \frac{1}{\log_2 n}, & n > 1 \\ 1, & n = 1 \end{cases} \tag{3}$$

Where n is the number of relation clusters that p_i belongs to. If a pattern p_i only appear in one relation cluster, its clarity C_i achieve the maximum value 1.

Following the theory of DCM, the weight of pattern p_i within relation R, $W_{i,R}$, is defined as follows:

$$W_{i,R} = \frac{Sig_{i,R}^2 * C_i^2}{\sqrt{Sig_{i,R}^2 + C_i^2}} * \sqrt{2} \tag{4}$$

To filter out noisy patterns, we trade off significance and clarity to get the weight of patterns in a relation category following Equation 4. We remove the patterns with the weight less than θ through all of the relation category.

4 Experiments

We set up experiments to answer the following questions: (i) Does the proposed features improve the accuracy in comparison with the State-of-the-art? (ii) How does the pattern selection procedure affect the performance of system.

4.1 DataSet

Manually labeling training data is a time-consuming and labor intensive task. In this paper, DS is adopted to automatically generate the training data. We employ Wikipedia[3] as the target corpus and Freebase as the knowledge base.

In order to obtain the training data using DS, previous works usually abide by the following steps. First, the named entity recognition (NER) tagger segments textual data into sentences and finds entity mentions in the corpus. Then, the entity pairs are associated with Freebase RDF triples. If the entity pairs appear in Freebase, the relevant sentences are selected as positive examples. This data

[3] http://www.wikipedia.org/

generating procedure is simple and straightforward. However, it is inefficiency especially when processing large corpus. In this paper, we firstly use Lucene[4] to index the Freebase Wikipedia Extraction (WEX)[5]. Then, we extract Freebase RDF triples and query index with *subject+object* to find relevant sentences. Third, the sentences are filtered except in accordance with the following two conditions: (i) The maximum length of a sentence is L tokens. (ii) The gaps of entity pairs should not exceed G tokens. Finally, since classes with very few training instances are hard to learn, the relations with labeled samples exceed N are selected as our experimental data.

4.2 Experimental Settings

According to the definition in section 1, relation extraction can be seen as a multi-class classification problem and the task of relation extraction is to identify certain predefined relationship between two entities. Note that there might not exist any semantic relationship between the two entities.

In this paper, we mainly focus on the entity pairs contain some relationships. Because we do not attempt to filter out entity pairs with no relationships, all of the entity pairs generated in section 4.1 have certain relationship. The sentence length greatly influences dependency parse results and the smaller the gaps of two entities is, the more likely that they have some relations. In the following experiments, we fix the maximum length of a sentence $L = 40$ and the maximum gaps of entity pairs $G = 15$. We pay attention to the relations with labeled samples exceed 2000. For each relation, we randomly select 1000 examples as the training instances and 1000 example as test instances. Finally, six mostly frequently mentioned relations are preserved. These include, for example, *written_by*, *produced_by*, *place_of_birth*, *profession*,*directed_by* and *nationality.*

After selecting dataset, we then use method proposed in section 3 to extract feature vectors. Each feature vector is assigned a relational label according to the relation that exists between the two entities. Finally, we train a multi-class classifier to learn a classification model to classify all of the relation types. For simplicity, we usually use logistic regression as our classifier.

To evaluate the performance of proposed feature, the test instances automatically generated by DS are regarded as a gold standard. Once all the test instances have been classified, they can be ranked by confidence score and the precision@N is used to evaluate the topmost results by the classification model.

4.3 Our Proposed Feature vs. State-of-the-art

In DS relation extraction, multiple sentences corresponding to an entity pair may be achieved. How to leverage information from multiple sentences is crucial for the final performance especially when it is not sufficient to deduce the relationship between two entities only from one sentence. Mintz et al. [11] simply

[4] http://lucene.apache.org/

[5] http://wiki.freebase.com/wiki/WEX

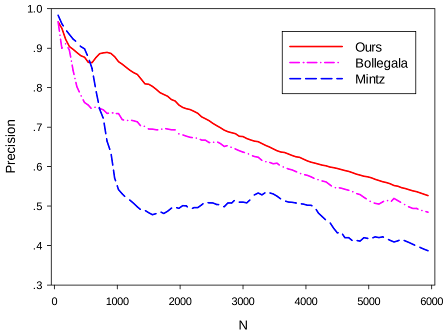


Fig. 3. The precision@N curve of six relation types.

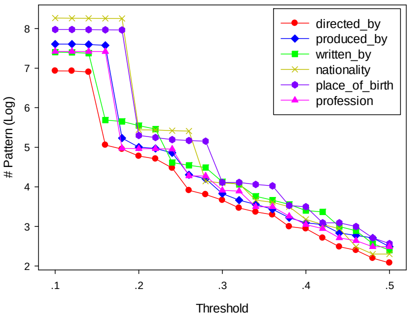
combine feature extracted from different sentences as a richer feature vector to deal with this problem. Besides, Bollegala et al. [4] proposed relation dual representation and entity pairs are represented as the distribution over patterns. Then, entity pairs are clustered to identify sematic relations. To evaluate the performance of this paper proposed feature, we select Mintz et al. [11] and Bollegala et al. [4] proposed feature and as competitors to compare with pattern space features.

The precision@N results are presented in Figure 3, *Ours* is the feature proposed in this paper and we select the best DCM selection threshold $\theta = 0.2$ (See section 4.4). In addition, we simply use L2-regularized multi-class logistic regression[6] as classifier for all of these features. From Figure 3, we can have the following observation: (i) *Mintz* proposed feature achieves relatively high precision with a small N and encounters a sharp decline in the precision when N is greater than 900. (ii) *Ours* achieves about 13% improvement over *Mintz* and 4% improvement over *Bollegala* at precision@6000. In the dataset, about 9.5% of the test instances encounter *null pattern* problem when using *Bollegala*. Pattern space features have made a better solution of the *null pattern* problem. *Ours* achieves the best result, which demonstrates the effectiveness of pattern space features.

4.4 The Effect of Pattern Selection

The number of patterns in the pattern space is the final dimension of feature vector. Excessive number of patterns not only leads to high computational cost but also feature sparsity. In this paper, we developed a pattern selection procedure to find the typical and discriminative patterns based on DCM. We experimentally study the effect of pattern selection procedure. First, we study the impact of DCM threshold value on the number of patterns. Figure 4 presents the changing curve of the pattern number along with DCM threshold θ. We can see that the

[6] http://www.csie.ntu.edu.tw/ cjlin/liblinear/



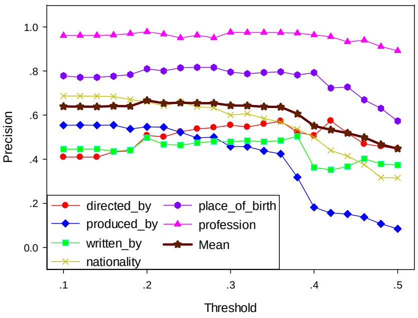


Fig. 4. The sensitivity of pattern number against DCM threshold.

Fig. 5. The effect of varying the DCM threshold.

pattern number suffers a sharp decline when the threshold between 0.1 and 0.2. Next, we investigate the effect of DCM threshold value on the average precision of various relation types. Figure 5 presents the precision of six types against the threshold θ. *Mean* represents the mean average precision of six relation types. We can see that the precision reaches the maximum value when the threshold is set $\theta = 0.2$, which is the best parameter that used in the above experiments. At the same time, there is no significant difference about the number of the patterns belonging to each relation type. Consequently, the mean average precision reaches the maximum value is reasonable. It is apparent that the pattern selection algorithm can select distinctive basis patterns and greatly reduce the dimensionality of feature vector while maintaining the high precision.

5 Conclusion

We proposed pattern space features for relation extraction. Pattern space features can leverage information from multiple sentences to deduce the relations between two entities. Each dimension of pattern space feature vector corresponds to a basis pattern. For two pattern space feature vectors to match, all of their dimensions no longer need to match exactly due to pattern similarity function. Furthermore, to avoid noisy patterns, we devised a pattern selection procedure to filter out patterns with low distinctiveness. Experimental results demonstrate that the proposed feature significantly outperforms State-of-the-art. The proposed pattern filtering procedure are effective for the improvement of precision.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.61202329, 61272332, 61333018), CCF-Tencent Open Research Fund and the Opening Project of Beijing Key Laboratory of Internet Culture and Digital

Dissemination Research(ICDD201301). We thank the anonymous reviewers for their insightful comments.

References

1. Agichtein, E., Gravano, L.: Snowball: extracting relations from large plain-text collections. In: Proceedings of the fifth ACM conference on Digital libraries. pp. 85–94 (2000)
2. Akbik, A., Visengeriyeva, L., Herger, P., Hemsen, H., Löser, A.: Unsupervised discovery of relations and discriminative extraction patterns. In: Proceedings of the 24th International Conference on Computational Linguistics. pp. 17–32 (2012)
3. Bach, N., Badaskar, S.: A review of relation extraction. Literature review for Language and Statistics II (2007)
4. Bollegala, D.T., Matsuo, Y., Ishizuka, M.: Relational duality: unsupervised extraction of semantic relations between entities on the web. In: Proceedings of the 19th international conference on World wide web. pp. 151–160 (2010)
5. Brin, S.: Extracting patterns and relations from the world wide web. In: Selected papers from the International Workshop on The World Wide Web and Databases. pp. 172–183 (1999)
6. Bunescu, R., Mooney, R.: Subsequence kernels for relation extraction. Advances in neural information processing systems 18, 171 (2006)
7. Bunescu, R.C., Mooney, R.J.: A shortest path dependency kernel for relation extraction. In: Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing. pp. 724–731 (2005)
8. Fung, G.P.C., Yu, J.X., Lu, H.: Discriminative category matching: Efficient text classification for huge document collections. In: Data Mining, 2002. ICDM 2003. Proceedings. 2002 IEEE International Conference on. pp. 187–194 (2002)
9. Harris, Z.: Distributional structure. Word 10(23), 146–162 (1954)
10. Hasegawa, T., Sekine, S., Grishman, R.: Discovering relations among named entities from large corpora. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics (2004)
11. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2. pp. 1003–1011 (2009)
12. Qian, L., Zhou, G., Kong, F., Zhu, Q., Qian, P.: Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In: Proceedings of the 22nd International Conference on Computational Linguistics. pp. 697–704 (2008)
13. Riedel, S., Yao, L., McCallum, A.: Modeling relations and their mentions without labeled text. In: Proceedings of the 2010 European conference on Machine learning and knowledge discovery in databases: Part III. pp. 148–163 (2010)
14. Rosenfeld, B., Feldman, R.: Clustering for unsupervised relation identification. In: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management. pp. 411–418 (2007)
15. Suchanek, F.M., Ifrim, G., Weikum, G.: Combining linguistic and statistical analysis to extract relations from web documents. In: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 712–717 (2006)