Syntactic Representation Learning for Open Information Extraction on Web

Chengsen Ru, Jintao Tang, Shasha Li, Ting Wang
College of Computer, National University of Defense Technology
No.137, Yanwachi Street, Changsha, China
{ruchengsen, tangjintao, shashali, tingwang}@nudt.edu.cn

ABSTRACT

This paper proposes a representation learning based method to discover new relations between entities from web, which is more general than existing Open Information Extraction(OIE) methods. Given dependency sequences on the expandPath as input, a convolutional neural network(CNN) is adopted to learn the representation layer features of the syntactic dependency patterns which indicate the relations. Experimental results show that compared with the state-of-art OIE methods, the proposed method obviously improves recall without much expense of precision.

CCS Concepts

•**Information systems → Information extraction;**

Keywords

Representation learning; Dependency sequences; CNN; Relation discovery

1. INTRODUCTION

Open Information Extraction(OIE)[1] from the web is helpful for many web intelligence applications, i.e. question answer, knowledge graph and etc.. OIE techniques are domain independent and unlimited to pre-specified relations by using syntactic patterns. However, almost all the existing OIE systems only extract the relations whose syntactic representation are exactly matched with the limited syntactic patterns which are handcrafted or learned from the training data. Considering the scale of web data, the limited patterns in OIE are not general enough to cover various kinds of the relations, which usually results to low recall in OIE task.

In order to solve the generalization problem in OIE, we propose a representation learning based method which could capture the higher-level linguistic features on top of the syntactic dependency patterns. The proposed method uses the dependency sequences on the expandPath[4] as input and adopts a CNN model to learn the representation features of syntactic patterns which are more general to extract the relations. Our method fundamentally differs from previous methods in that it is able to discover the relations presented in new syntactic patterns.

©2017 International World Wide Web Conference Committee
(IW3C2), published under Creative Commons CC BY 4.0 License.
WWW 2017 Companion, April 3–7, 2017, Perth, Australia.
ACM 978-1-4503-4914-7/17/04.
http://dx.doi.org/10.1145/3041021.3054266

2. METHODOLOGY

2.1 Dependency Parsing

Dependency parsing provides a description of the directed syntactic relations between governor and dependent words in a sentence. The syntactic relations in dependency tree can yield simplified syntactic patterns of relations between entities which are widely used in OIE[3, 4], e.g., WOE generates its patterns by identifying the shortest dependency paths (called corePath) between entities[4].

However, as relations can be presented in various forms, the syntactic patterns are still too specific to recognize the broad relations on the web. On top of these patterns, it is reasonable to believe that there are more general syntactic characters to represent the relations, which might be useful to improve the generalization of relation discovery. It has been proved that CNN can be used to learn the representations of higher-level features from the raw data[5]. So this paper adopts a multilayer CNN model to learn the representations of abstract features on top of syntactic dependency trees by using the dependency sequences on the expandPath as input.

2.2 The Structure of CNN

The CNN model for OIE mainly includes three parts: dependency representation, feature extraction and output.

In dependency representation, in order to capture the dependency order, each dependency is transformed to a one-hot vector by looking up the dependency matrix $W_e \in R^d$, where d is the number of different dependency types and each dimension of W_e denotes a type of dependency. Each dependency sequence consists of dependency relations between a governor word and all of its dependent words on the expandPath. Each dependency sequence is represented by a concatenation of these one-hot vectors with respect to dependent word orders in the source sentence. The fixed size of the concatenation vector is d_s, where $d_s = d \times l$ and l is the maximum length of all dependency sequences. Lastly, the dependency sequences of a sentence are represented by a matrix $X \in R^{d_s \times t}$, where t is the number of governor words on the expandPath.

In feature extraction, the network uses two convolutional layers and a max pooling layer to extract features. The first convolutional layer uses a linear transformation $W_1 \in R^{n_1 \times d_s}$ to extract local feature from the matrix X, where n_1 is the number of the convolutional kernels in the first convolutional layer. The resulting matrix Z has size of $n_1 \times t$.

$$Z = W_1 X + b_1 \tag{1}$$

In order to capture the most useful local features produced by the first convolutional layer, we perform a max pooling over Z to produce a global feature vector Z_m, which has a fixed size of n_1, independent of the number of the input dependency sequences.

The second convolutional layer uses hyperbolic *tanh* as the non-linearity function to extract more meaningful features. $W_2 \in R^{n_2 \times n_1}$ is the linear transformation matrix, where n_2 is the number of the convolutional kernels in the second convolutional layer.

$$Z_f = tanh(W_2 Z_m + b_2) \tag{2}$$

The output vector Z_f of the second convolutional layer is then fed to a softmax classifier to compute the confidence of whether there is a relation among the entity pair or not. The softmax classifier is used to predict a 2-class distribution d(x) and the transformation matrix is $W_3 \in R^{2 \times n_2}$. We denote $t(x) \in R^{2 \times 1}$ as the target distribution vector: the entry $t_k(x)$ is the probability whether there is a relation between the entity pair or not.

To learn the network parameters, we optimize the cross-entropy error between $d(x)$ and $t(x)$ using stochastic gradient descent(SGD). Given all training instances, we define the objective as:

$$J(\theta) = -\sum_{x} \sum_{k=1}^{2} t_k(x) log(d_k(x)) \tag{3}$$

where θ represents the parameters need to learn. Gradients are computed using back propagation. We minimize the log likelihood $J(\theta)$. W_1, b_1, W_2, b_2 and W_3 are randomly initialized.

3. EXPERIMENTS AND RESULTS

The proposed method was evaluated on a dataset obtained by randomly selecting 50,000 articles from English Wikipedia. By collecting all possible triples[4] in these articles, we got a candidate test set consisting of 1,095,894 triples. We randomly selected 1000 triples from the set: 500 triples for validation, 500 triples for test. All triples were manually labeled as positive (expressing a relation) or negative (expressing no relation) by 3 evaluators and then voted to the final label.

SemEval-2010 Task 8 dataset[2] contains 10,717 annotated entity relation examples, all of which can be classified into 9 specific relationship classes and the *Other* class. In our experiments, the instances of the *Other* class were removed and the others were used as positive training data. In order to get the negative training data, another 500 triples were randomly selected from the candidate test set and labeled in the same way as the test set. The negative labeled triples formed the negative training data.

We tuned the hyper parameters on the validation set for each experimental setting. The hyper parameters include l, n_1 and n_2. The best setting was obtained with the values: 7, 200 and 100 respectively.

Table 1: The performance of methods in discovering relations

Method	Precision	Recall	F1
DECNN	0.765	**0.711**	**0.737**
TECNN	0.749	0.556	0.638
OLLie	**0.792**	0.560	0.656

We compared the following 3 methods: DECNN(proposed in this paper), TECNN[5](the state-of-art method of using CNN for relation-specific, which took terms on the corePath as input) and OLLie[3](the state-of-art method of OIE, which generated extraction patterns from dependency parsing). We evaluated all these methods by precision, recall and F1.

Table 1 summarizes the performance of the methods. Compared with other methods, while keeping the comparative precision, DECNN obviously improved the recall and F1 value. For details, in the true relations discovered by DECNN, 35.3% dependency patterns were new. The results demonstrated that based on the syntactic patterns, CNN model has learned more general and higher-level features from input dependency sequences, which could be used to recognize new patterns indicating relations.

4. CONCLUSIONS

In this paper, we focus on automatically exploiting more general features than syntactic patterns to represent the relations in OIE. A CNN based method is proposed to learn the representations of higher-level features from syntactic dependency sequences. The experimental results show that our method can effectively discover the relations no matter presented in known or new syntactic patterns.

5. ACKNOWLEDGMENTS

This work was sponsored by the National Natural Science Foundation of China (No.61472436, 61532001 and 61303190).

6. REFERENCES

[1] M. Banko, M. J. Cafarella, S. Soderland, et al. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[2] I. Hendrickx, S. N. Kim, Z. Kozareva, et al. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *SemEval*, pages 33–38, 2010.

[3] M. Schmitz, R. Bart, S. Soderland, et al. Open language learning for information extraction. In *EMNLP*, pages 523–534, 2012.

[4] F. Wu and D. S. Weld. Open information extraction using wikipedia. In *ACL*, pages 118–127, 2010.

[5] K. Xu, Y. Feng, S. Huang, and D. Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *EMNLP*, pages 536–540, 2015.