SPAN BASED OPEN INFORMATION EXTRACTION

Junlang Zhan
Department of Computer Science and Engineering
Shanghai Jiao Tong University
longmr.zhan@sjtu.edu.cn

Hai Zhao *
Department of Computer Science and Engineering
Shanghai Jiao Tong University
zhaohai@cs.sjtu.edu.cn

ABSTRACT

In this paper, we propose a span based model combined with syntactic information for n-ary open information extraction. The advantage of span model is that it can leverage span level features, which is difficult in token based BIO tagging methods. We also improve the previous bootstrap method to construct training corpus. Experiments show that our model outperforms previous open information extraction systems. Our code and data are publicly available at https://github.com/zhanjunlang/Span_OIE

1 Introduction

Open Information Extraction (Open IE) aims to generate a structured representation of information from natural language text in the form of verbs (or verbal phrases) and their arguments. An Open IE system is usually domain independence and does not rely on pre-defined ontology schema. For example, give a sentence "*repeat customers can purchase luxury items at reduced prices*", the extraction can be (*repeat customers; can purchase; luxury items; at reduced prices*). Open IE have been widely applied in many downstream tasks such as textual entailment [1] and question answering [2].

The first Open IE system is TextRunner [3]. A number of improvements were made by the following systems: Reverb [4] improved the relation extraction by leveraging POS tag patterns; OLLIE [5] built a large training corpus by using ReVerb and alignment to database and then extracted rules by using syntactic pattern; ClausIE [6] built elaborate rules using syntactic information; Stanford Open IE [7] split complex sentences into simple sentences before extracting information; OpenIE4 [8] was developed based on OLLIE and Semantic Role Labeling (SRL) [9]; OpenIE5 was an advanced version of OpenIE4 which improved the extraction of numerical sentences [10] and conjunctive sentences [11]; Graphene [12] parsed a sentence into tree structure structure before doing the extraction. Most of these systems use unsupervised or semi-supervised approaches and rule-based algorithms due to the lack of high quality labeled data.

Stanovsky et al. [13] firstly created a labled corpus by an automatic translation from question-answer driven semantic role labeling (QA-SRL) annotations [14] and they developed an Open IE system PropS using a BiLSTM labeler and BIO tagging scheme. Cui et al. [15] constructed a large noisy corpus by applying OpenIE4 and selected tuples with high confidence score. They also built an Open IE system by using neural sequence to sequence model and copy mechanism to generate extractions. However, this system can only perform binary extractions.

In this paper, we developed an Open IE system by adapting a modified span selection model which was applied in SRL [16] and by applying span level syntactic information. Span model is also applied in other fields such as coreference resolution [21]. The advantage of span model is that span level features can be exploited which can not be performed in BIO tagging models. We also constructed a large training corpus following the method of Cui et al.. The differences of our construction method and the method of Cui et al. are two-fold: firstly, our corpus is constructed for n-ary extraction instead of binary extraction. Secondly, previous method only use extractions with high confidence scores for training.

*Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), Key Projects of National Natural Science Foundation of China (U1836222 and 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

However, the confidence scores estimated by OpenIE4 are not highly reliable and we found that some extractions with low confidence scores contain important information. So we proposed a method which can leverage also information of extraction with low confidence scores. Experiments show that our system outperforms previous Open IE systems.

2 Model

2.1 Problem Definition

We treat Open IE as a span selection problem. The goal is to select correct span for each label. Given a sentence $W = w_1, ..., w_n$, we wish to predict a set of predicate-argument relations $Y \subseteq S \times P \times L$, where $S = \{(w_i, ..., w_j) | 1 \leq i \leq j \leq n\}$ is all the spans in W, P is the set of predicate spans which is a subset of S and $L = \{A0, A1, A2, A3\}$ is the label set of Open IE. For each label, we wish to select a span (i, j) that has the highest score:

$$\arg\max_{(i',j') \in S} SCORE_l(i', j'), l \in L$$

where

$$SCORE_l(i, j) = P_\theta(i, j|l) = \frac{exp(\phi_\theta(i, j, l))}{\sum_{(i',j') \in S} exp(\phi_\theta(i', j', l))}$$

and ϕ_θ is a trainable scoring function with parameters θ. To train the parameters θ, in the training set, for each sample X and the gold structure Y^*, we minimize the cross-entropy function:

$$l_\theta(X, Y^*) = \sum_{(i,j,l) \in Y^*} -logP_\theta(i, j|l)$$

Note that some labels may not appear in the given sentence and predicate span. In this case, we define the predicate span as NULL span and train a model to select the NULL span when the label dose not appear in the sentence.

2.2 Model Architecture

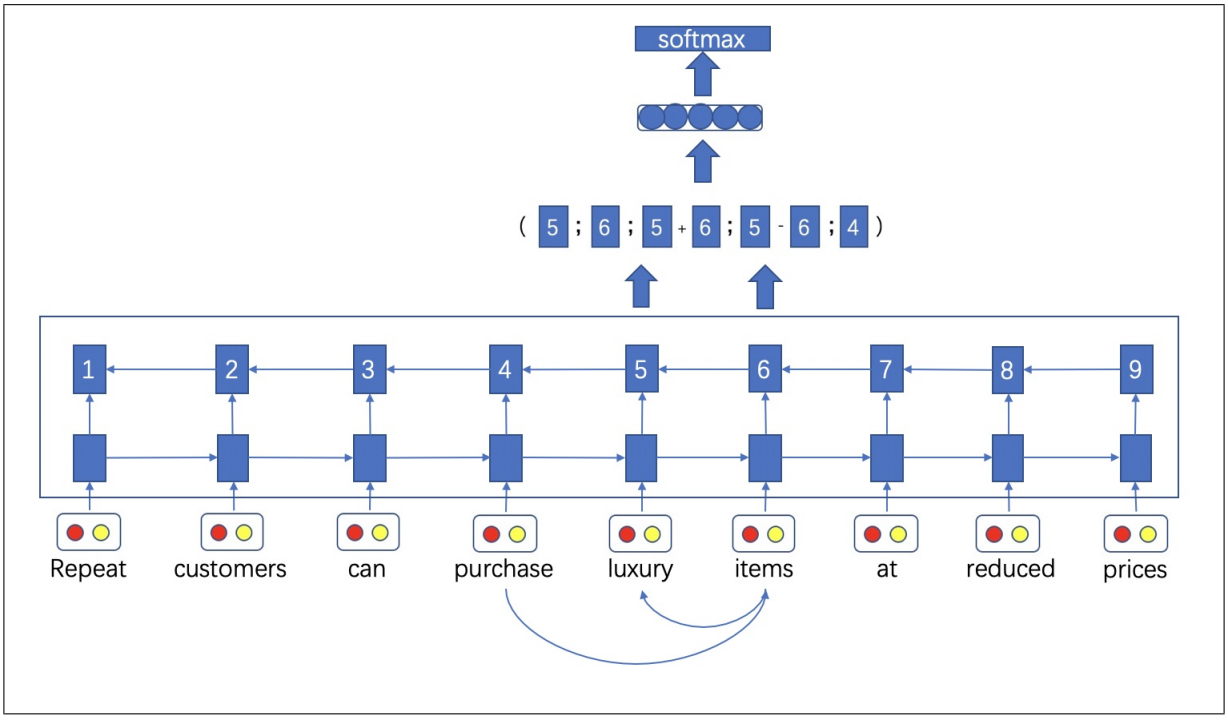

Figure 1: An overview of our model

To calculate the score of each span, we use a neural network combined with syntactic information which is shown in Figure 1. Our model, named Span OIE, is a BiLSTM-Span model, inspired by the state of the art deep learning

approach for SRL suggested by Ouchi et al. [16]. Given an input sample which contains a sentence S and a predicate span P, we extract a feature vector x_i for every word $w_i \in S$:

$$x_i = emb(w_i) \oplus emb(pos(w_i)) \oplus emb(p(w_i)) \oplus emb(dp(w_i))$$

where $emb(w_i)$ is the word embedding, $emb(pos(w_i))$ is the pos tag embedding, $emb(p(w_i))$ is the embedding for indicator of predicate which is a binary value, $emb(dp(w_i))$ is the embedding of the type of dependency parsing, which proves its effectiveness in the task of sentence compression [18] and $\oplus$ denotes concatenation.

The word vectors are then fed into a BiLSTM [19] which computes contextualized output features:

$$h_i = BiLSTM(x_i)$$

To leverage span features, we create span features as followed:

$$f_{span}(s_{i:j}) = h_i \oplus h_j \oplus h_i + h_j \oplus hi - h_j \oplus h_{dp(s_{i:j})}$$

where $s_{i:j}$ is the span which starts from w_i and ends at w_j. h_i and h_i are used as a part of span features in the work of He et al. [20]. $h_i + h_j$ and $h_i - h_j$ are used as span features in the work of Ouchi et al [16]. $dp(s_{i:j})$ is the parent word of syntactic head of the span $s_{i:j}$. In the example of Figure 1, the syntactic head of the span *[luxury items]* is the word *luxury*, whose parent is the word *purchase*. So h_4 is also a part of the span features in this example. We call this syntax information as span level syntax information. We believe this information works because all the words are not equally important in the same span. For example, in the span *[luxury items]*, the syntax information of the word *items* is more important than that of the word *luxury*. The span features are then fed into a feed forward network and a softmax layer to obtain the scores of different labels for a span. In the inference stage, we use span-consistent greedy search proposed by Ouchi et al [16].

2.3 Predicate Model

The model above needs the predicate span as input. To obtain the predicate spans, we also train another span model which is similar to the model above. The differences are two-fold. Firstly, we remove the span level syntax information. Secondly, the feed forward network becomes a binary classification network to judge whether a span is predicate or not. In the inference stage, if a predicate is completely included in another predicate in the same sentence, we drop this predicate. For example, given sentence <*James agreed to sell his company*>. If two span *[agreed to sell]* and *[to sell]* are both selected as predicate, we keep only the former one. So our system is a pipeline. We firstly find the predicates for a sentence using predicate model, then we feed every predicate into the span model to obtain extractions.

2.4 Confidence Score

Like other Open IE systems, our model also provide a confidence score for every extraction. The confidence score of an extraction is defined as followed:

$$confidence\_score = score_{pred} \times \sum_{i=1}^{label\_number} score_{i,span_i}$$

where $score_{pred}$ is given by the softmax value of the predicate model, $score_{i,span_i}$ is the score of selected span for label i given by the span model.

3 Data

We use a part of raw corpus provided by Cui et al. [15][2] and OpenIE4[3] to do the n-ary extraction. Although there exists OpenIE5[4] which is an advanced version of OpenIE4, we still use OpenIE4 because the improvements of OpenIE5 are the extractions for numerical information and for conjunctive sentences by breaking conjunctions in arguments to generate multiple extractions. The extraction of OpenIE4 and OpenIE5 are the same in most cases. However, OpenIE5 takes much more time while performing extractions. In the work of Cui et al. [15], they filter the extraction with the confidence score greater than 0.9. However, we observe that the confidence score provided by OpenIE4 is not highly reliable. For example, in the sentence *he makes a state visit.*, The extraction by OpenIE4 is *[he;makes;a state visit]*. This extraction looks quite correct while the confidence score is 0.388 out of 1. We further observe that all

[2] `https://1drv.ms/u/s!ApPZxTWwibImHl49ZBwxOU0ktHv`

[3] `https://github.com/allenai/openie-standalone`

[4] `https://github.com/dair-iitd/OpenIE-standalone`

the extractions that contains *he, she, they, it* are under low scores. Since Open IE is a task of information extraction. Pronoun seems contain not such information. However, we argue that extraction with pronoun is still useful. For example, if the sentence *he makes a state visit* is in the article about president Obama, we can easily infer that *he* refers to president Obama so this sentence still contain useful information. We also find that there exists other extractions which look quite correct but under low scores. So abandoning all the extractions under 0.9 will lose some important information. To deal with this issue, we improve the loss function as followed:

$$l_\theta(X, Y^*) = confidence\_score(Y^*) \times \sum_{(i,j,l) \in Y^*} -logP_\theta(i, j|l)$$

That is to say, we leverage all the extractions obtained by OpenIE4 but for every extractions, we multiply its loss by its confidence score. In this way, we can obtain information from correct extractions with low confidence score and for the truly bad extractions, this coefficient can also lower their influences.

4 Experiments

4.1 Spans Candidates Selection

The main drawback of span model is that the number of the spans is usually too large which will cause a large computational cost and hurt the performance of the model. Supposed the length of a sentence is T, the number of possible spans is $\frac{T(T+1)}{2}$. To reduce the number of span, we propose 3 constraints for span candidates:

- **Length constraint**: We keep only the spans with length less than 10.
- **No overlap constraint**: We keep only the spans which are not overlap with the predicate span.
- **Syntactic constraint**: We keep only the spans which satisfy that a word is either the syntactic parent of another word in the same span, or the parent of this word is in the same span. For the example shown in Figure 1, the span *[luxury items at]* is not a qualified span because the word *at* is not the parent of a word in the same span and its parent, the word *purchase* is not in this span.

4.2 Experiment Settings

In the experiments, we use glove [5] with dimension 100 as word embedding. We use spaCy [6] to perform the dependency parsing. The dimension of hidden state of BiLSTM is set as 200, the number of layers of BiLSTM is 2. The dimension of POS tag embedding is 10 and the dimension of embedding of dependency parsing label is 20. We train the model for 100000 steps and the batch size is set as 20. We use Adam as optimizer and the initial learning rate is set as 0.01. The decay rate is set as 0.001 for every 100 steps.

4.3 Results

We used the benchmark and scripts constructed by Stanovsky et al. [17][7] to evaluate the precision and recall of different Open IE systems. The precisio-recall (P-R) curve and Area under P-R curve (AUC) are shown in Figure 2. We can observe that the Span OIE system outperforms all other tested systems and the syntactic information further improves the performance of the base model. Although our model is trained from the corpus bootstrapped from the output of OpenIE4, it still have a better performance than OpenIE4 because it mainly use the extractions with high confidence scores. The improvement of our model compared with OpenIE4 is two-fold. Firstly, our model can find more predicates than OpenIE4 which leads to more extractions thus higher recall. Secondly, our model are more accurate in finding the correct arguments. As an example shown in Table 1, our model finds one more predicate <*to access*> and the correct arg0 <*The keys*>.

5 Conclusion, Limitations and Future Works

In this paper, we proposed a neural span based open information extraction system. The model is trained with corpus bootstrapped from outputs of OpenIE4 system. We improved the method of construction of training corpus which

[5] https://nlp.stanford.edu/projects/glove/

[6] https://spacy.io

[7] https://github.com/gabrielStanovsky/oie-benchmark

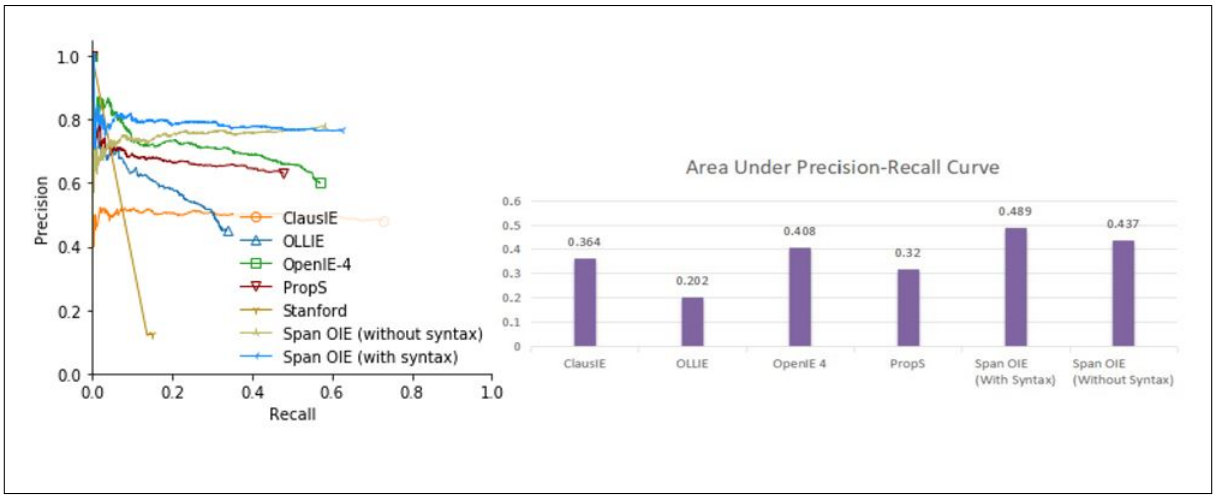

Figure 2: The P-R curve and AUC of Open IE systems

Table 1: Example of extractions from Span OIE and OpenIE 4

Orginal sentence	The keys, which are needed to access the building, were locked in the car.
Span OIE	<The keys ; were needed ; to access the building> <The keys ; **to access** ; the building> <**The keys** ; were locked ; in the car>
OpenIE 4	<The keys ; were needed ; to access the building> <The keys, which were needed ; were locked ; in the car>

can leverage useful information in extractions with low confidence scores. Our system outperforms previous Open IE systems in both precision and recall. However, our system still has limitations. Firstly, some previous Open IE systems like OLLIE, OpenIE4 and ClausIE also extract context of a sentence. For example, given a sentence *<He believes The Lakers will win the game.>* The correct extraction should be *context<He believes>; The Lakers; will win ; the game* because *<The Lakers will win the game>* is not a fact. That is also an important difference between Open IE and SRL since Open IE need to identify correct and useful information. The reason why we did not bootstrap context information is that the context information extracted by OpenIE 4 is very noisy. Another type of extraction that our model can not perform is appositive. For example, given a sentence *<Obama, president of USA, gave a speech on Friday.>*, one extraction should be *<Obama;[be];president of USA>*. These two challenges are also not tackled in other appeared neural Open IE systems. Besides these two challenges, a better way to construct training corpus for Open IE is still under research. At last, more effective span features need to be explored to further improve the model.

References

[1] Berant, J., Dagan, I., & Goldberger, J. (2011). Global Learning of Typed Entailment Rules. ACL.

[2] Fader, A., Zettlemoyer, L.S., & Etzioni, O. (2014). Open question answering over curated and extracted knowledge bases. KDD.

[3] Yates, A., Banko, M., Broadhead, M.G., Cafarella, M.J., Etzioni, O., & Soderland, S. (2007). TextRunner: Open Information Extraction on the Web. HLT-NAACL.

[4] Fader, A., Soderland, S., & Etzioni, O. (2011). Identifying Relations for Open Information Extraction. EMNLP.

[5] Mausam, Schmitz, M., Soderland, S., Bart, R., & Etzioni, O. (2012). Open Language Learning for Information Extraction. EMNLP-CoNLL.

[6] Corro, L.D., & Gemulla, R. (2013). ClausIE: clause-based open information extraction. WWW.

[7] Angeli, G., Premkumar, M.J., & Manning, C.D. (2015). Leveraging Linguistic Structure For Open Domain Information Extraction. ACL.

[8] Mausam (2016). Open Information Extraction Systems and Downstream Applications. IJCAI.

[9] Christensen, J., Mausam, Soderland, S., & Etzioni, O. (2011). An analysis of open information extraction based on semantic role labeling. K-CAP.

[10] Saha, S., Pal, H., & Mausam (2017). Bootstrapping for Numerical Open IE. ACL.

[11] Saha, S., & Mausam (2018). Open Information Extraction from Conjunctive Sentences. COLING.

[12] Cetto, M., Niklaus, C., Freitas, A., & Handschuh, S. (2018). Graphene: A Context-Preserving Open Information Extraction System. COLING.

[13] Stanovsky, G., Michael, J., Zettlemoyer, L.S., & Dagan, I. (2018). Supervised Open Information Extraction. NAACL-HLT.

[14] He, L., Lewis, M., & Zettlemoyer, L.S. (2015). Question-Answer Driven Semantic Role Labeling: Using Natural Language to Annotate Natural Language. EMNLP.

[15] Cui, L., Wei, F., & Zhou, M. (2018). Neural Open Information Extraction. ACL.

[16] Ouchi, H., Shindo, H., & Matsumoto, Y. (2018). A Span Selection Model for Semantic Role Labeling. EMNLP.

[17] Stanovsky, G., & Dagan, I. (2016). Creating a Large Benchmark for Open Information Extraction. EMNLP.

[18] Wang, L., Jiang, J., Chieu, H.L., Ong, C.H., Song, D., & Liao, L. (2017). Can Syntax Help? Improving an LSTM-based Sentence Compression Model for New Domains. ACL.

[19] Graves, A., Fernández, S., & Schmidhuber, J. (2005). Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. ICANN.

[20] He, L., Lee, K., Levy, O., & Zettlemoyer, L.S. (2018). Jointly Predicting Predicates and Arguments in Neural Semantic Role Labeling. ACL.

[21] Lee, K., He, L., Lewis, M., & Zettlemoyer, L.S. (2017). End-to-end Neural Coreference Resolution. EMNLP.