Open Event Trigger Recognition Using Distant Supervision with Hierarchical Self-attentive Neural Network

Xinmiao Pei[1], Hao Wang[1,2](✉), Xiangfeng Luo[1,2](✉), and Jianqi Gao[1]

[1] School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China
{peixinmiao,wang-hao,luoxf,gjqss}@shu.edu.cn

[2] Shanghai Institute for Advanced Communication and Data Science, Shanghai 200444, China

Abstract. Event trigger recognition plays a crucial role in open-domain event extraction. To address issues of prior work on restricted domains and constraint types of events, so as to enable robust open event trigger recognition for various domains. In this paper, we propose a novel distantly supervised framework of event trigger extraction regardless of domains. This framework consists of three components: a trigger synonym generator, a synonym set scorer and an open trigger classifier. Given the specific knowledge bases, the trigger synonym generator generates high-quality synonym sets to train the remaining components. We employ distant supervision to produce instances of event trigger, then organizes them into fine-grained synonym sets. Inspired by recent deep metric learning, we also propose a novel neural method named hierarchical self-attentive neural network (HiSNN) to score the quality of generated synonym sets. Experimental results on three datasets (including two cross-domain datasets) demonstrate the superior of our proposal compared to the state-of-the-art approaches.

Keywords: Open-domain event extraction · Synonym set · Distant supervision · Neural network

1 Introduction

In event extraction, the discovery of unseen event triggers is one of the important and challenging tasks. It is also a fundamental pre-processing step for a wide range of natural language processing (NLP) applications, such as text summarization [8], question answering [3] and text mining [4]. Compared to a large number of studies carried out in mining entity synonyms, there is less attention that has been devoted to event trigger synonym extraction.

Problem Definition. Given an unstructured raw text data D, the task of event trigger recognition aims to discover all event trigger instances E then organize these triggers into synonym sets S so that each set is corresponding to

H. Yang et al. (Eds.): ICONIP 2020, CCIS 1332, pp. 701–708, 2020.
https://doi.org/10.1007/978-3-030-63820-7_80

one latent event type. However, open-domain synonym set generation of event trigger is challenging since determining the proper granularity of event synonym sets underlying different latent event types is difficult. Despite its importance, event trigger synonym extraction in the open domain remains an under-explored problem.

A straightforward way is supervised learning an event trigger detector. Prior studies mainly treat it as a multi-classification problem. Liu and Shulin [7] present a method to encode argument information in event detection via supervised attention mechanisms. Recently, Nguyen [9] and Sha [11] exploit syntactic information to enhance event detection. These models are typically trained on a small size of the human-annotated dataset in closed domains and thus subject to over-fitting.

However, rare research has been conducted for open event type identification. Most existing methods employ different graph clustering algorithms [2,10,13]. The problem is divided into two categories: (1) ranking methods based on similarity, and (2) detecting and organizing. However, this approach ignores intra-association among event triggers which may benefit the quality of discovered trigger synonym sets. Shen and Jiaming [12] attempts to model the interactions across entity synonym sets. This method is not suitable for event trigger synonym extraction where event triggers are usually domain-specified.

To address these issues, we introduce a novel framework based on distant supervision paradigm. Our method can be split into three phases: training instance and synonym set generation, synonym set quality evaluation and open event trigger classification. Our goal is to detect all probable event triggers and latent types without a specified event type list. Our contribution is as follows:

– We introduce a new framework of open event trigger extraction, whose goal is to detect all probable existing event triggers and latent types simultaneously.
– Human annotation of events in the open domain is substantially expensive. This method provides a universal solution to generate high-quality event trigger instances using distant supervision.
– A novel model, called HiSNN (Hierarchical Self-attentive Neural Network), is proposed as the score function for measuring intra-distance metric considering the hierarchical semantic relationship between trigger elements and trigger instances.

2 Our Model

We firstly give formal definitions for the existing notions in this paper. An event **trigger instance** is a phrasal representation, which includes the context information including the key event trigger word e and the complementary participant or destination c. Following the above definition, e and c are valid **trigger elements**.

This framework consists of three components: (1) a trigger synonym generator, which matches the unstructured raw text in the given knowledge base to event trigger instances; (2) a synonym set scorer, this phase build a neural model to score the trigger synonym set; and (3) an open trigger classifier to determine whether an unseen event trigger instance should be inserted into the existing event trigger synonym set or a new empty set.

2.1 Dataset Generation Using Distant Supervision

Given some known structural event instances, distant supervision provides a simple and effective way to discover more event triggers that are not in the knowledge base from raw text corpus by masking the trigger words. To automatically extend the annotation coverage, we assume that if a trigger synonym exists in one sentence, then this sentence is likely to be an event mention. For each event type existing in the base, we construct a synonym dictionary, so that we can successfully query unseen trigger words out of the pre-defined vocabulary of trigger words. As a result, we extract the training examples from the training data required for the next phase of open event type recognition.

Example 1. Given a sentence "Jia Yueting solved the problem of arrears of listed companies by selling assets to obtain funds.","A company conducts asset auctions through judicial procedures." an entity linker may first map "asset sale" and "assent auction". Then, we found the trigger words "capital for debt payment", expanded from knowledge base. Finally, we get an trigger synonym set"asset sale", "assent auction","capital for debt payment".

2.2 Trigger Synonym Set Scorer Architecture

In this subsection, we mainly explain how to score all trigger instances that are known to belong to a specific trigger synonym set. The right of Fig. 1 shows a high-level overview of the approach. The model is composed of four modules: 1) The input layer consists of two parts: a trigger synonym set and its corresponding trigger elements. 2) Instance representation learns instance feature in the trigger synonym set. 3) Element Representation extracts the corresponding element from instances in the known trigger synonym set. 4) Scorer layer makes a final representation of the synonym score.

Input Layer. The input consists of two parts: a trigger synonym set $S = \{s_1, s_2, s_3, \ldots, s_n\}$ and its corresponding trigger elements $W = \{w_1, w_2, w_3, \ldots, w_m\}$. For the trigger elements, we denote the dimension of element embeddings by $m \times R^d$, where m is the number of elements. For the trigger synonym set, we concatenate all argument vectors as the representation of the whole trigger instance, so the set embedding is represented by $l \times 5 \times R^d$, where l is the maximum number of instances in the set.

Instance Representation. We use a fully connected neural network with two hidden layers to extract the overall characteristics of trigger instances. Then, we sum all instance features and obtain the raw instance representation $v(S) =_{i=1}^{n} \phi(s_i)$.

Element Representation. The goal of this module is to produce the element represented. We employ bidirectional LSTM (BiLSTM), which consists of both

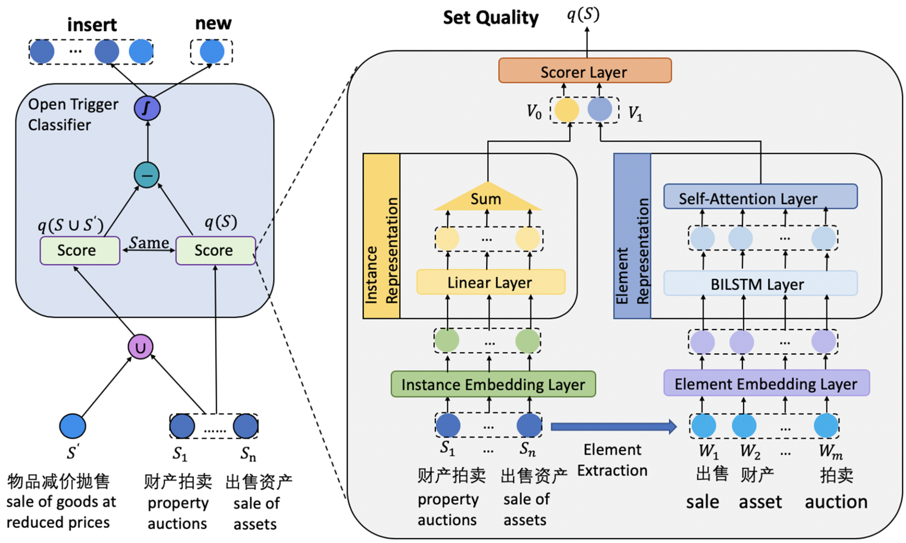

Fig. 1. Model architecture of the open event trigger classifier.

forward and backward networks to process the set. Then, we use the attention mechanism to learn dependence within the element and capture the internal structure of elements. The attention mechanism takes the wohle $LSTM$ hidden state H as input, and outputs a vector of weights $v(W)$.

$$\overrightarrow{h_t} = \overrightarrow{\mathrm{LSTM}}(\overrightarrow{f_t}, \overrightarrow{h_{t-1}}) \tag{1}$$

$$\overleftarrow{h_t} = \overleftarrow{\mathrm{LSTM}}(\overleftarrow{f_t}, \overleftarrow{h_{t+1}}) \tag{2}$$

$$H = [\overrightarrow{h_t}, \overleftarrow{h_t}] \tag{3}$$

$$v(W) = softmax(W_{s_2} tanh(W_{s_1} H)) \tag{4}$$

Scorer Layer. We first concatenate instance representation $v(S)$ and element representation $v(W)$ to obtain final set representation, then, we construct it using another fully connected neural network with three hidden layers to get our set score.

2.3 Learning Open Trigger Classifier

The right part of Fig. 1 shows a set scorer $q(\cdot)$ which takes a known trigger synonym set S as input, and returns a quality score S that measures how complete and coherent this set S is. Given this set S and an unseen trigger instance s', our open trigger classifier $f(S, s')$ first applies the set scorer to obtain input set score $S(i.e., q(S))$. Then, we add the unseen instance s' into the set and apply the set scorer again to obtain the quality score of $S \cup s'$. Finally, we calculate the difference between these two quality scores, and transform this score difference into the probability using a sigmoid unit as follows:

$$Pr(s' \in S) = f(S, s') = \phi(q(S \cup s') - q(S)) \tag{5}$$

Given a collection of k trigger set-instance pair $(S_i, s'_i)|_{i=1}^k$ with their corresponding labels $y_i|_{i=1}^k$, we learn the open trigger classifier using the log-loss as follows:

$$L(f) = \sum_{i=1}^{m} -y_i \log(f(S_i, s'_i)) - (1 - y_i) \log(1 - f((S_i, s'_i))) \tag{6}$$

where y'_i equals to 1 if $s' \in S$ equals to 0 otherwise.

2.4 Open Event Trigger Synonym Set Generation Algorithm

Inspired by [6], we design open event trigger synonym set generation algorithm for mine all event trigger set. This algorithm takes the above learned open trigger classifier model, all unseen triggers $T = s_1, s_2, s_3, \ldots, s_{|T|}$, and a probability threshold θ as input, and clusters all unseen triggers into trigger synonym sets. Specifically, this algorithm enumerates all unseen event trigger T once and maintains a pool of all detected synonym sets E. For each unseen trigger $s_i \in T$, it applies the open trigger classifier f to calculate the probability of adding this trigger into each detected set in E and finds the best trigger synonym set E_j that has the largest probability. If this probability value passes the threshold θ, and we will add s_i into set E_j. Otherwise, we create a new event trigger synonym set s_i with this single trigger and add it into the set S. The entire algorithm stops after one pass of the unseen triggers and returns all detected trigger synonym sets E. Note that we do not need to specify the number of sets, and our open event trigger synonym set generation algorithm will determine this value on its own. In this work, we simply set the probability threshold of θ to be 0.5 and study its Influence on clustering performance below.

3 Experiment

3.1 Datasets

In order to verify our model, we conduct our experiment on three datasets. The first dataset is financial domain[1], which is an event ontology corpus. It contains 303,363 news documents, including 2,648 event instances. To make the experiment more universal and validate the effectiveness of our proposal on cross-domain datasets, we construct two additional datasets based on THUCNews[2]. One dataset is in education and social domain which contains 278,649 news documents including 2,384 event instances, and the other dataset is in the entertaining and political domain which contains 186,102 news documents, including 1,845 events instances.

1 https://github.com/Meow0727/Finance/upload.
2 http://thuctc.thunlp.org/.

Table 1. Quantitative results of open event trigger recognition. We run all methods ten times and calculated their average score.

Method	Finance			Education&Society			Entertainment&Politics		
	ARI	FMI	NMI	ARI	FMI	NMI	ARI	FMI	NMI
Kmeans	27.62	27.88	84.26	43.66	44.95	86.01	40.91	43.10	83.81
Louvain [1]	31.74	32.15	88.64	40.31	43.98	81.27	42.07	52.11	83.59
Cop-Kmeans [14]	36.53	40.27	80.19	48.86	56.49	87.30	50.28	51.58	86.72
SVM+Louvain	9.59	14.53	69.27	8.86	9.37	51.33	12.98	16.80	70.27
L2C [5]	12.87	19.90	73.47	12.71	16.66	70.23	7.76	8.79	70.08
SynSetMine [12]	57.14	59.15	88.88	55.41	57.03	88.74	58.65	61.56	89.00
HiSNN	**64.77**	**65.41**	**88.28**	**62.79**	**65.57**	**89.99**	**65.86**	**67.28**	**90.25**
BERT-Instance	53.02	58.05	88.46	60.91	64.81	90.42	58.17	63.65	90.26
BERT-Element	67.26	69.87	91.75	74.99	76.27	93.26	75.38	78.00	93.52
BERT-HiSNN	**69.53**	**71.16**	**91.48**	**84.00**	**84.96**	**95.27**	**83.52**	**84.31**	**95.22**

3.2 Experimental Settings

For a fair comparison, we utilize pre-trained 50-dimensional word embedding vectors on different domain datasets. For HiSNN, we use a neural network with two hidden layers (of sizes 50, 250) as liner layer and another neural network with three hidden layers (of sizes 250, 500, 250) as scorer layer (c.f. Fig. 1). We optimize our model using Adam with an initial learning rate of 0.001 and apply dropout technique with a dropout rate of 0.3. For the generation algorithm, we set the probability threshold of θ to be 0.5. Following previous work [12], we measure ARI (Adjusted Rand Index), FMI (Fowlkes mallows score) and NMI (Normalized mutual information) to evaluate the results.

3.3 Results and Analysis

We first compare the performance of all methods for classifying event instances. As shown in Table 1, the results demonstrate that our method achieves an ARI of 64.77,FMI of 65.41 and NMI of 88.28 in finance domain. The main disadvantage of unsupervised methods (Kmeans and Louvain) and semi-supervised mothod(Cop-Kmeans) is that they cannot utilize supervision signals. The deficiency of SVM+Louvain and L2C does not have a holistic view of identifying event type due to they base on pairwise similarity. For SynSetMine, the relations between the elements is ignored. Therefore, our model HiSNN can capture event type-level features beyond pairwise similarity, besides, we take into account the importance among elements in determining event type. Furthermore, We utilize BERT pre-training vectors, we can see that the overall model performs better than BEAT-Instance and BEAT-Element. Besides, BERT-HiSNN performs better on the cross-domain dataset compared to the single-domain dataset due to BERT is more universal.

Then, we conduct more experiments to analyze each component of our HiSNN model in more details and show some detailed case studies.

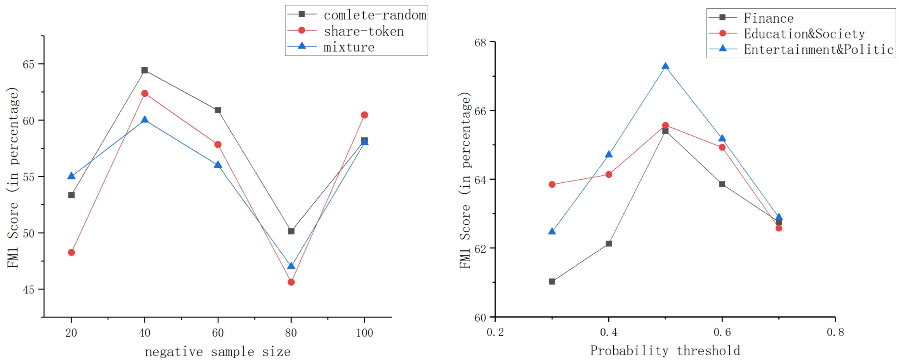

Fig. 2. Hyper-parameters analysis.

On the one hand, in order to train our classifier, we need to convert all event instances generated from distant supervision into event set-instance pair. Therefore,we study how different conversion strategy and negative sample sizes affect model performance on financial datasets. Results are shown in Fig. 2. We find that the complete-random strategy actually performs better than the share-token strategy easily. One possible explanation is that the complete-random strategy can generate more diverse negative samples and thus provide more supervision signals. On the other hand, in order to verify the influence of different thresholds on the algorithm, intuitively, the higher this threshold θ is, the more conservative our algorithm will be, and more event types will be generated. Therefore, we run our algorithm with fixed our model and different thresholds. The results are shown in Fig. 2. In all the above experiments, we set the threshold θ to be 0.5. we notice that the performance of our clustering algorithm is insensitive to θ and value within 0.4 and 0.6 is generally good for θ. Second, we find that setting θ to be 0.5 is robust and works well across all three cross-domain datasets.

4 Conclusion

This paper investigated mining unseen event triggers using distant supervision and distance metric learning. We presented a novel framework that effectively leverages Chinese knowledge bases to generate the training instances, then learn an open event trigger classifier to score trigger synonym sets, which allows detecting new event triggers and event types. We also proposed a hierarchical self-attentive neural network to score trigger synonym sets. In the future work, we will try to further automatically integrate prior knowledge into instance generation using scalable neural variational inference and learn this model in an end-to-end fashion.

Acknowledgment. The research reported in this paper was supported in part by the Natural Science Foundation of China under the grant No.91746203, National Natural

Science Foundation of China under the grant No.61991415, and Ministry of Industry and Information Technology project of the Intelligent Ship Situation Awareness System under the grant No. MC-201920-X01.

References

1. Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. J. Stat. Mech.: Theory Exp. **2008**(10), P10008 (2008)
2. Cybulska, A., Vossen, P.: Translating granularity of event slots into features for event coreference resolution. In: Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation, pp. 1–10 (2015)
3. Feng, W., Wu, Y., Wu, W., Li, Z., Zhou, M.: Beihang-msra at semeval-2017 task 3: a ranking system with neural matching features for community question answering. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), pp. 280–286 (2017)
4. Ferracane, E., Marshall, I., Wallace, B.C., Erk, K.: Leveraging coreference to identify arms in medical abstracts: an experimental study. In: Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis, pp. 86–95 (2016)
5. Hsu, Y.C., Lv, Z., Kira, Z.: Learning to cluster in order to transfer across domains and tasks. arXiv preprint arXiv:1711.10125 (2017)
6. Kumar, A., Chatterjee, S., Rai, P.: Nonparametric bayesian structure adaptation for continual learning. arXiv preprint arXiv:1912.03624 (2019)
7. Liu, S., et al.: Exploiting argument information to improve event detection via supervised attention mechanisms (2017)
8. Nallapati, R., et al.: Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv preprint arXiv:1602.06023 (2016)
9. Nguyen, T.H., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: Thirty-second AAAI Conference on Artificial Intelligence (2018)
10. Oliveira, H.G., Gomes, P.: ECO and onto. PT: a flexible approach for creating a portuguese wordnet automatically. Lang. Resour. Eval. 48(2), 373–393 (2014)
11. Sha, L., Qian, F., Chang, B., Sui, Z.: Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
12. Shen, J., Lyu, R., Ren, X., Vanni, M., Sadler, B., Han, J.: Mining entity synonyms with efficient neural set generation. Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 249–256 (2019)
13. Ustalov, D., Panchenko, A., Biemann, C.: Watset: Automatic induction of synsets from a graph of synonyms. arXiv preprint arXiv:1704.07157 (2017)
14. Wagstaff, K., Cardie, C., Rogers, S., Schrödl, S., et al.: Constrained k-means clustering with background knowledge. In: Icml, vol. 1, pp. 577–584 (2001)