



Event Relation Reasoning Based on Event Knowledge Graph

Tingting Tang¹, Wei Liu^{1(✉)}, Weimin Li¹, Jinliang Wu², and Haiyang Ren²

¹ School of Computer Engineering and Science, Shanghai University,
Shanghai, China
liuw@shu.edu.cn

² The 54th Research Institute of China Electronics Technology Group,
Shijiazhuang, China

Abstract. Natural language text contains numerous event-based, and a large number of semantic relations exist between events. Event relations express the event rationality logic and reveal the evolution process of events, which is of great significance for machines to understand the text and the construction of event-based knowledge base. Event relation discovery includes extracting event relation from text and obtaining event relation by reasoning. Event relation extraction focuses on the recognition of explicit relations, while event relation reasoning can also discover implicit relations, which is more meaningful and more difficult. In this paper, we propose a model combining LSTM and attention mechanism for event relation reasoning, which uses the attention mechanism to dynamically generate event sequence representation according to the type of relation and predicts the event relation. The macro-F1 value in the experimental result reaches 63.71%, which shows that the model can effectively discover implicit event-event relation.

Keywords: Event knowledge graph · Event relation · Bi-LSTM · Attention mechanism

1 Introduction

Events and event relations in natural language contain advanced semantic information. Events do not occur in isolation, and the occurrence of the event is logically related to other events. The event relation is different from the classification relation between traditional concepts. It is often used to describe the higher-level semantic relations between events, such as causality, follow, concurrency, and composite. Discover event relations from text helps machines understand text better, and facilitate the construction of event-based knowledge bases from text.

Supported by the National Key Research and Development Program of China (No. 2017YFE0117500), the National Natural Science Foundation of China (No. 61991410), the research project of the 54th Research Institute of China Electronics Technology Group (No. SKX192010019).

Event relation discovery includes event relation extraction and event relation reason. Recent research work on event relation extraction mainly aims at temporal event relation detection [17], subevent relation recognition [1, 10], and causality relation extraction [7]. These methods are mainly divided into pattern matching method and machine learning method. The pattern matching method generally uses the rule template to match the keywords of the relation, and the machine learning method constructs the model to capture the semantics and features of the text for relation extraction. Existing methods focus on the extraction of explicit relations and cannot predict event relations from event-event sequence, while event relation reasoning can also discover implicit relations, which is more meaningful and difficult.

In this paper, we aim at reasoning implicit relations of events from text. Event relation reason is a task to obtain the pairs of event sequences and classify the relation (causality, follow, concurrency, and composite) between them. Existing methods of event relation discovery based on machine learning generally use models such as RNN and GCN [5, 6, 21] to obtain the representation of the event. However, the text contains redundant information unrelated to the event, resulting in the weakness of semantic information in the event feature vector. Event knowledge graph is event-based knowledge base for specific application domain. It is usually constructed by domain experts iteratively through manual or automatic methods. Event knowledge graph describes important events with related elements (such as action, objects, time and place) in a specific domain in the form of RDF triples, and also contains the description of rationality relationship between events. So we can generate the event representation sequence from it and take it as a priori knowledge base for event relation reasoning.

In this paper, we propose a model combining LSTM and attention mechanism for event relation reasoning. We obtain event information from the event knowledge graph and use the attention mechanism to dynamically generate event sequence representation according to the type of relation. To verify the effectiveness of the proposed event relational reasoning method, we annotate a new text set about COVID-19 on the basis of CEC¹ corpus, and construct an event knowledge graph about COVID-19 from the annotated texts.

The remainder of this paper is organized as follows: Sect. 2 introduces and discusses relevant works about the recognition model of different event relations. Section 3 introduces the event knowledge graph. Section 4 describes our proposed model. Section 5 demonstrates relevant experiments and analyzes the results. Finally, Sect. 6 concludes by summarizing our proposed method and pointing out directions for future work.

2 Related Work

Recent research work on event relation extraction tasks mainly aims at causality relation extraction, temporal relation detection and subevent relation recognition.

¹ <https://github.com/daselab/CEC-Corpus>.

Causality relation is the most important semantic relation between events, Girju [7] constructed a template to identify causality relations and used it to match the keywords of the relation. Peng [18] proposed a method to measure causality on event triggers by using pointwise mutual information. Recently, Mirza [14] presented a data-driven approach with rules to extract causal relations between events. Kriengkrai [9] extracted the original sentence of the causal candidate from the network text and the multi-column CNN is used to determine whether there is a causal relation between event pairs.

Event temporal relations specify how different events in a paragraph relate to each other in terms of time sequence. Current work aimed at representing event pairs based on linguistic features and using statistical learning methods (such as logistic regression [12] and SVM [15]) to capture the relations. With the development of deep learning technology, Cheng [5] and Xu [21] extracted the shortest dependency path of the event context and classified the event temporal relation with neural network based on LSTM. Dai [6] combined LSTM and GCN to capture features and correlation syntax for event temporal relation detection.

Besides, some research aims at the extraction of event hierarchical relations. This task attempts to extract hierarchical structure where the parent event contains child events described in the same document. To cope with this task, Araki [2] introduced a multi-class logistic regression model to detect subevent relations. Glavaš [8] constructed rich set of features for subevent relation classification. Zhou [24] constructed a common temporal sense language model Tacolm and predicted subevent relations on this basis.

Existing research works obtain event features and representations from text and uses different models to learn a particular relation between events, individually. These methods mainly aim at the extraction of explicit relations of texts, while a large number of semantic relations are implicit. Moreover, they are primarily developed at the sentence level and hence it is easy to omit relational event pairs scattered in different sentences or even in different documents.

In this paper, we construct the event knowledge graph and generate the event sequence based on it. Then we use a neural network model to learn the relation characteristics between events and predict the event relation.

3 Event Knowledge Graph

Based on the event model and event ontology concept proposed by Liu [11], we propose the event knowledge graph. Event knowledge graph is an event-based knowledge base accumulated in knowledge application, which contains event-based knowledge in different fields.

To construct the event knowledge graph, from the perspective of knowledge representation, we define an event as

$$E := \langle A, P, O, L, T \rangle \quad (1)$$

where A, P, O, L, T represents the trigger, participant, object, location, and time of the event, respectively. The event knowledge graph includes event ontology models and event instances, which is described as

$$EKG := \langle EO_s, EI_s \rangle \tag{2}$$

where EO_s is event ontology set and EI_s is event instance set. The event ontology is a shared, formal and explicit specification of the event class system model. It contains event class concepts, rules for event knowledge inference and relations between event classes (including causality, follow, concurrency, and composite).

An event instance is an instantiation of an event class, representing a specific event. We extend the RDF triples to represent the basic elements in the event knowledge graph. For example, the relation between the event instance and the event class can be described as $\langle e_1, type, ec_1 \rangle$, where e_1 is an event instance and ec_1 is an event class. Event relations describe the rationality relation between events, represented as $\langle e_1, event_relation, e_2 \rangle$. Event elements is the description of the event, which is defined as: $\langle e_1, element_relation, element_entity \rangle$, where $element_relation$ is selected from $\{hasParticipant, hasObject, hasLocation, hasTime\}$ and $element_entity$ represents the event element entity.

Event knowledge graph is usually constructed by domain experts iteratively through manual or automatic methods. Machine learning methods are widely used in the automatic construction of event knowledge graph, such as GNN [16] is used to extract event trigger, and K-means is used to identify event elements.

Proof. “Since the COVID-19 epidemic spread in Europe and the world, the European Commission will continue to restrict travel to Europe from mid-March to 15 June this year.” The event knowledge graph from the text is shown in Fig. 1.

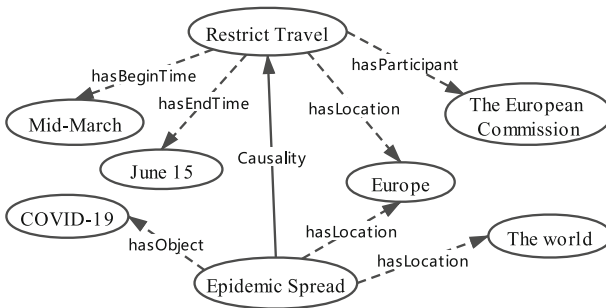


Fig. 1. A simple subgraph of the event knowledge graph.

4 Proposed Method

In this model, word2vec is used to map element words into low-dimensional vectors space, and LSTM is used to model event sequence. In addition, Attention mechanism assigns different weights to event element vectors.

4.1 Event Representation

In this paper, the event sequence is generated from the event knowledge graph. In CEC, we find that the average length of event sequence is 4.85, and 96.63% of event elements (trigger, participant, object, location and time) can be covered completely when the length of the event sequence is twelve. So we set the length of the event sequence as twelve. In a Chinese sentence, events are usually described in the order of time, participants, triggers, objects, and place. Therefore, we obtain the event trigger and related elements from the event knowledge graph in order. The event sequence contains six words filled forward and five words filled backward in the above order based on the event trigger. We use “<pad>” to mark paddings to ensure that the length of event sequence is equal.

4.2 LSTM

We use LSTM, a special RNN, to process sequence data and learn long-term dependencies. As shown in Fig. 2, the LSTM layer runs on a vector sequence of event sequences. LSTM has three gates (input i , forget f and output o) and a cell storage vector. The forget gate decides what information to throw away from the cell state. The input gate determines how the input vector $x(t)$ changes the cell state. The output gate allows the cell state to affect the output.

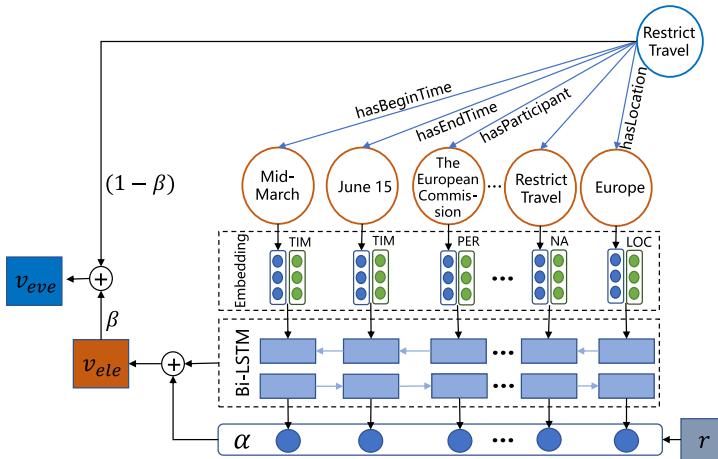


Fig. 2. The event-event relation reason model based on Bi-LSTM and attention mechanism. The input consists of two parts: an event sequence with NER tags and a target event-event relation. v_{ele} and v_{eve} are embedding vectors of the event and event element. The weight vector α is calculated based on the target event-event relation.

4.3 Word Encoder

Word embeddings learned from a large amount of unlabeled data can capture the semantics of words. In this paper, we use word2vec [13] to learn word embeddings on the corpus. Considering the effect of event element type on event representation, we concatenate the word vector e_{word} and the type vector e_{type} to represent each word.

$$w_t = e_{word} \oplus e_{type} \quad (3)$$

As the reversed order of event sequence also carries rich information, we can feed the reverse event sequence into LSTM to obtain the backward representation. Bidirectional LSTM uses two LSTMs to obtain the semantics of the forward and backward sequences, respectively. Therefore, for an event represented by sequence w_t , $t \in [0, T]$, the calculation is described simply as follows:

$$\vec{h}_t = \overrightarrow{LSTM}(w_t), t \in [1, T] \quad (4)$$

$$\overleftarrow{h}_t = \overleftarrow{LSTM}(w_t), t \in [T, 1] \quad (5)$$

We concatenate the forward hidden state \vec{h}_t and the backward hidden state \overleftarrow{h}_t to contain the semantics of the event, which is described as $event_i = [\vec{h}_t, \overleftarrow{h}_t]$.

4.4 Attention Layer

Each event is usually triggered by a verb and described by several event elements. For event pairs with different relations, the focus of the event elements is different. For example, when reasoning whether the event pair has a concurrent relation, we will emphasize two events' time continuity. Attention Mechanism [3] can guide the neural network model to treat each component of the input unequally according to the importance of the input to a given task. So, we use the attention mechanism to measure the importance of event elements in relation type and assign different weights to the elements.

In Fig. 2, the attention weight α is calculated from the hidden state h calculated by Bi-LSTM and the target event relation vector r . The attention score of the k th event element word in the event sequence is calculated as:

$$\alpha_k = \frac{\exp(\mathbf{h}_k \cdot \mathbf{r}^T)}{\sum_i \exp(\mathbf{h}_i \cdot \mathbf{r}^T)} \quad (6)$$

$$v_{element} = \alpha^T \mathbf{H} \quad (7)$$

where $\mathbf{H} = [\mathbf{h}_1, \dots, \mathbf{h}_T]$ is a matrix and $v_{element}$ is the vector of event elements which integrates all event element of the event. The vector of event sequence is defined as the weighted sum of $v_{trigger}$ and $v_{element}$, which is described as

$$v_{event} = (1 - \beta) \cdot v_{trigger} + \beta \cdot v_{element} \quad (8)$$

where $\beta \in [0, 1]$ trades off event trigger and elements and $v_{trigger}$ is the hidden vector of the trigger word token in Bi-LSTM.

4.5 Event-Event Relation Reason

Inspired by the translation model in the knowledge graph, such as TransE [4] and TransH [20], we regard the event as a special entity and assume that the distance of event-event with relation in vector space is close. Considering that events and relations are in different semantic spaces, we map the event vector into the relation semantic space through the relation matrix. The score of event pairs under the relation is calculated as follows:

$$g(event_1, r, event_2) = \|W_r v_{event_1} + r - W_r v_{event_2}\| \quad (9)$$

The objective is to minimize the maximum interval loss function. The main idea of this method is that we should maximize the score interval between each positive sample $T_i = (event_1, r, event_2)$ and the negative sample T'_i generated by replacing an event in the triple randomly. The loss function is expressed as

$$J = \sum_{T_i \in S} \sum_{T'_i \in S'} \max(0, 1 - g(T_i) + g(T'_i)) \quad (10)$$

where S and S' represent the set of positive and negative examples, respectively.

5 Experiments

In this section, we present the experiments on event relation reason. Specifically, we conduct evaluation for event relation extraction based on CEC corpus (Sect. 5.1–Sect. 5.4). We demonstrate the effects of event elements on event relation reasoning by adjusting the parameter β (Sect. 5.5). Finally, a detail ablation study is given to explain the significance of attention mechanism (Sect. 5.6).

5.1 Dataset

In experiments, we use CEC, a Chinese event-based dataset, to generate the event knowledge graph. Figure. 3 shows the annotation content of corpus by taking the event *travel restriction to Europe* as an example. Besides, we use 92 representative reports about COVID-19 and mark the event triggers, event elements (such as participants, time, location), and event relations semi-automatically. We add 3459 event instances and 1616 event relations to the CEC in total.

5.2 Experimental Settings

The dimension of the word vector is 128 dimensions, which word2vec trains. The vector of event element type is a 32-dimensional vector initialized randomly. The size of the LSTM is 128 (256 for Bi-LSTM), and the number of layers in the LSTM is 2. During the training process, we used gradient descent and the Adam optimizer algorithm to update the parameters. The batch size is 64. The learning rate is 0.02, and we use the exponential slowing method to make the learning rate decrease exponentially with the increase of the training step. To prevent neural networks from over-fitting, we adopt dropout [19] and L2 regularization.

```

<Event eid="e5">
  <Participant sid="s5" type="agent">欧盟(the European Commission)
</Participant>将继续(will continue to)
  <Denoter did="d5" type="operation">限制(restrict)</Denoter>赴
  <Location tid="t5">欧洲(Europe)</Location>
  <Object oid="o5">旅行(travel)</Object>, 从(from)
  <Time tid="t5" type="relTime">今年3月中旬(mid-March this year)</Time>至(to)
  <Time tid="t5" type="relTime">6月15日(15 June)</Time>
</Event>

```

Fig. 3. The annotation of *travel restriction to Europe* event in the corpus. For ease of reading, the English translation of the Chinese text is added in the brackets.

5.3 Evaluation Metrics

We use precision (P), recall (R) and F_1 -measure (F_1) to evaluate the results.

Precision: the percentage of correctly predicted relations in total predicted event relations.

Recall: the proportion of correctly predicted event relations in the relation samples.

$$F_1\text{-measure: } F_1 = \frac{2 * P * R}{P + R}$$

5.4 Result and Discussion

To verify the effectiveness of the model, we compare with the following methods:

- **word+PoS+SVM:** For the event sequence, we spliced word vectors based on the bag-of-words model and part of speech tags as event features and used SVM as a classifier.
- **word+PoS+KNN:** We use the same feature engineering above but use KNN as the classifier.
- **Yang:** Yang et al. [22] proposed an event-based Siamese Bi-LSTM network to model events and applied it to the learning of causality.
- **Zhang:** Zhang [23] used deep LSTM with average pooling in the last layer to obtain the average vector value and learn the event temporal relation.

Table 1. Performance comparison with different methods

Method	Causality	Follow	Concurrency	Composite	Overall
word+PoS+SVM	61.67	38.93	30.98	50.45	45.51
word+PoS+KNN	64.89	46.44	34.67	57.84	50.96
Yang's model	70.59	42.82	41.19	69.21	55.95
Zhang's model	71.16	45.96	39.79	72.50	57.35
our model	73.18	50.75	48.16	82.76	63.71

Table 1 reveals the performance comparison of the five models on the dataset. The macro-average F_1 value of our method is 63.71%. Our model achieves better

performance on the relations of *causality* and *composite*, compared with *follow* and *concurrency*. There are two possible reasons. Firstly, event pairs with *causality* and *composite* relations have a strong logical connection with event triggers, and their distribution is relatively concentrated. Secondly, the *follow* and *concurrency* relation focus on the time element of event pair. However, the model does not distinguish well between two relations.

From Table 1, we can find that the overall performance of our model is better than the four benchmarks in macro-average F_1 value. Compared with the machine learning-based methods, our neural network model significantly improves the F_1 value. These results further verify the effectiveness of the neural network in the task of event relation reason.

The comparison of model performances on individual relation types also proves the improvement on the F_1 score. Our model achieves the greatest performance improvement (72.5 vs. 82.76) on *composite* relation. The relations of *follow* and *concurrency* are so difficult to distinguish that the traditional methods cannot recognize them well. However, the BiLSTM can capture the semantic features of event sequences, and the attention mechanism assigns different weights to event elements. Hence, our model significantly improves the performance of two relations. These results prove the validity of our model.

5.5 The Effect of β

Table 2 lists the detailed performance of our model with different parameters β . We select numbers from $\{0, 0.1, 0.2, 0.3, 0.4\}$ as the parameter β to verify the effect of event elements on event relation reason. From the table, we can find that it is better to add event elements to the event representation than to take the event trigger’s hidden vector as the event’s representation. When β is 0.2, the model is most suitable for the event relation reason task and achieves the best performance. When β is greater than 0.2, the model becomes less effective. It is likely that the lack of event trigger information and the emphasis on event element information resulting in insufficient semantic coherence of the event. These results indicate that integrating event trigger and elements to represent event semantic information is more suitable for event-event relation reason.

Table 2. The effect of β on experimental performance

β	Precision (%)	Recall (%)	F1-score (%)
0.0	60.27	61.25	60.65
0.1	62.23	62.91	62.52
0.2	64.05	63.74	63.71
0.3	61.54	61.90	61.46
0.4	60.40	61.22	60.69

5.6 Analysis of Attention Weight α

We design two sets of comparative experiments to confirm the impact of the attention mechanism. In the first set of experiments, we use the attention mechanism to assign different weights to event elements in a specific relation and take the weighted summation result as the event’s element representation. In the second set of experiments, we only add the average vectors of event elements. The experimental results are shown in Fig. 4. In general, the model combining the attention mechanism has a better experimental result. The attention mechanism can dynamically assign weights to elements according to the type of relation and capture the features of specific event elements.

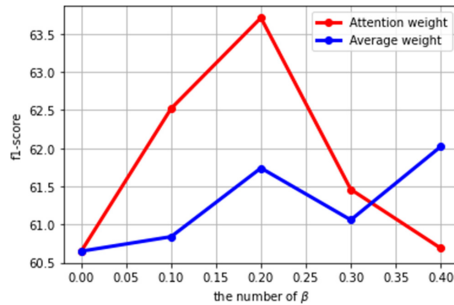


Fig. 4. Influence of attention mechanism. **Red line:** the representation of event elements obtained by the attention mechanism. **Blue line:** the representation of the event element obtained by adding the average element vectors. (Color figure online)

Figure 5 shows several examples of the attention vector α learned by our model. In the first example, the *causality* relation focuses on the logical connection between event pairs, and the model assigns similar attention scores to event elements. The time element and the continuity of the event occurrence are important clues to the *follow* relation. In the second example, the model focuses not only on time but also on the continuity of the event pairs. Therefore it assigns high attention scores to key elements of the event, such as “fire” and “fireman”. Furthermore, the *concurrency* relation refers to the occurrence of two events simultaneously or one after the other. The third example is a negative sample. However, to explore the *concurrency* relation between events, the model tries to capture this feature by assigning a high attention score to the time element.

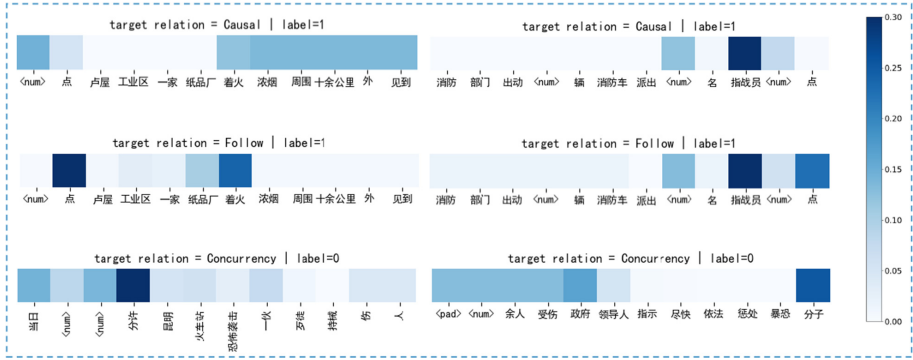


Fig. 5. Visualization of attention weight vector α of sample event instances learned by our model.

6 Conclusion

In this paper, we proposed an event relation reason model based on LSTM and attention mechanism. The event knowledge graph is introduced as a priori knowledge base and we obtain the event sequence from it. The model learns features for relation reasoning iteratively along the event representation sequence. We leverage LSTM for event information propagation and integration. Meanwhile, attention mechanism assigns different weights to event elements dynamically. Experimental results show that the model achieves a better performance on the reasoning of event causality and composition relation. In future work, we will improve the reasoning of concurrency and follow relation between events.

References

1. Aldawsari, M., Finlayson, M.A.: Detecting subevents using discourse and narrative features. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (2019)
2. Araki, J., Liu, Z., Hovy, E.H., Mitamura, T.: Detecting subevent structure for event coreference resolution. In: LREC, pp. 4553–4558 (2014)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint [arXiv:1409.0473](https://arxiv.org/abs/1409.0473) (2014)
4. Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Neural Information Processing Systems (NIPS) pp. 1–9 (2013)
5. Cheng, F., Miyao, Y.: Classifying temporal relations by bidirectional LSTM over dependency paths. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), pp. 1–6 (2017)
6. Dai, Q., Kong, F., Dai, Q.: Event temporal relation classification based on graph convolutional networks. In: Tang, J., Kan, M.-Y., Zhao, D., Li, S., Zan, H. (eds.) NLPCC 2019. LNCS (LNAI), vol. 11839, pp. 393–403. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32236-6_35

7. Girju, R., Moldovan, D.: Mining answers for causation questions. In: Proceedings of the AAAI Spring Symposium, October 2002
8. Glavaš, G., Šnajder, J.: Constructing coherent event hierarchies from news stories. In: Proceedings of TextGraphs-9: the Workshop on Graph-Based Methods for Natural Language Processing, pp. 34–38 (2014)
9. Kruengkrai, C., Torisawa, K., Hashimoto, C., Kloetzer, J., Oh, J.H., Tanaka, M.: Improving event causality recognition with multiple background knowledge sources using multi-column convolutional neural networks. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI 2017, p. 3466–3473. AAAI Press (2017)
10. Liu, Z., Mitamura, T., Hovy, E.: Graph-based decoding for event sequencing and coreference resolution. arXiv preprint [arXiv:1806.05099](https://arxiv.org/abs/1806.05099) (2018)
11. Liu, Z., Huang, M., Zhou, W., Zhong, Z., Fu, J., Shan, J., Zhi, H.: Research on event-oriented ontology model. *Comput. Sci.* **36**(11), 189–192 (2009)
12. Mani, I., Verhagen, M., Wellner, B., Lee, C., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics, pp. 753–760 (2006)
13. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint [arXiv:1301.3781](https://arxiv.org/abs/1301.3781) (2013)
14. Mirza, P.: Extracting temporal and causal relations between events. In: Proceedings of the ACL 2014 Student Research Workshop, pp. 10–17 (2014)
15. Mirza, P., Tonelli, S.: Classifying temporal relations with simple features. In: Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 308–317 (2014)
16. Nguyen, T., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
17. Ning, Q., Subramanian, S., Roth, D.: An improved neural baseline for temporal relation extraction. arXiv preprint [arXiv:1909.00429](https://arxiv.org/abs/1909.00429) (2019)
18. Peng, H., Song, Y., Roth, D.: Event detection and co-reference with minimal supervision. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 392–402 (2016)
19. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**(1), 1929–1958 (2014)
20. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge graph embedding by translating on hyperplanes. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 28 (2014)
21. Xu, Y., Mou, L., Li, G., Chen, Y., Peng, H., Jin, Z.: Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1785–1794 (2015)
22. Yang, Z., Liu, W., Liu, Z.: Event causality identification by modeling events and relation embedding. In: Cheng, L., Leung, A.C.S., Ozawa, S. (eds.) *ICONIP 2018*. LNCS, vol. 11303, pp. 59–68. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-04182-3_6

23. Zhang, Y., Li, P., Zhou, G.: Classifying temporal relations between events by deep bilstm. In: 2018 International Conference on Asian Language Processing (IALP), pp. 267–272. IEEE (2018)
24. Zhou, B., Ning, Q., Khashabi, D., Roth, D.: Temporal common sense acquisition with minimal supervision. arXiv preprint [arXiv:2005.04304](https://arxiv.org/abs/2005.04304) (2020)