Event Logic Graph Construction for Event Mining

Lei Li¹, Xiaodong Shi¹, Yang Ding¹, Yicheng Sun¹, Dong Han¹, Jingyang Li²,*

¹The 28th Research Institute of CETC, Nanjing 210007, China
²School of Computer Science and Technology, Xidian University, Xi’an 710071, China

*Corresponding author e-mail: jylee@stu.xidian.edu.cn

Abstract. Event Logic Graph is a directed and cyclic graph, the nodes in the graph represent events, and the edges represent the logical relationship between events. In essence, Event Logic Graph is a knowledge base of event logic, in order to reveal the evolution law and development mode of the event, we research a method of constructing Event Logic Graph, which describes the logical structural relationship between events by adopting knowledge graph structure. At the same time, in order to describe the event more clearly, we also describe the multi-dimensional attributes of the event. We propose an architecture for constructing Event Logic Graph, which includes text corpus collection, event relationship template construction, event extraction and structured representation, event similarity calculation and fusion, event trigger word extraction and argument extraction model construction, event relationship pair construction, graph database storage, and the architecture is used to construct Serial Event Logic Graph, Causal Event Logic Graph, Conditional Event Logic Graph, Transitional Event Logic Graph, and Concurrent Event Logic Graph.

Keywords: Event Logic Graph, Event Extraction, Event fusion, Event Relationship Extraction Template

1. Introduction
Event is one of the core concepts of human society, and people's social activities are often event-driven. The evolutionary laws and patterns that occur one after another in time and space between events are very valuable knowledge [1]. However, the existing typical knowledge graphs all focus on entities and their attributes and relationships, and lack the description of event logic, which is an important human knowledge. In order to make up for this shortcoming, Event Logic Graph [2] comes into being, which can reveal the evolutionary laws and development logic of events, portray and record human behavior. The Event Logic Graph describes the knowledge in a logical society, nodes represent events and edges between nodes represent logical relationships [3].

Human memory of the world is formed by the combination of multiple events. Events are more relevant to human thinking and behavior, and can better reflect the form and content of knowledge in the real world, especially the dynamic representation of knowledge. The existing knowledge graph cannot answer questions about event logic, such as "why" and "what to do next". However, many artificial intelligence applications currently require a deep understanding and use of event knowledge.
For example, for consumer intention recognition, the machine needs to know that there are often multiple consumer events in the "travel" event, such as "buy air tickets" and "book a hotel". For the dialogue system, it is difficult for the existing dialogue system to deeply understand causality of the dialogue, and the system need to understand common sense event knowledge, such as "taking a taxi" before "going to the airport", which makes system give smarter responses based on different contexts. For the financial field, studying the causal relationship between the rise and fall of the stock market is a valuable research content. For example, "rising food price" leads to "inflation", which leads to "falling stock". Knowledge of causality [4] like this play an important role in practical applications.

This paper uses natural language processing related theories and technologies to propose construction method and architecture of the Event Logic Graph. By using the knowledge graph structure to describe the logical structure relationship between events, at the same time, in order to describe the event more clearly, we also describe the multi-dimensional attributes of the event. The content of Event Logic Graph construction includes text corpus collection, event relationship extraction template construction, event extraction and structuring, event similarity calculation and fusion, event trigger word extraction and argument extraction model construction, event relationship pair construction, and graph database storage. We use this architecture to construct Serial Event Logic Graph, Causal Event Logic Graph, Conditional Event Logic Graph, Transitional Event Logic Graph, and Concurrent Event Logic Graph. Each graph is a directed and cyclic graph, with nodes representing events and edges representing event logical relations between events, such as causality, succession, conditions, and transitions.

2. Related Work

The related research work of Event Logic Graph was launched in 2014. GORAN et al. first proposed the concept of Event Graph [5]. Later, Rospocher et al.[6] proposed an event-centric knowledge graph in 2016, and the relationship between events is co-reference and causality. In 2019, Gottschalk et al.[7] proposed an event-centric temporal knowledge graph (Event-Centric Temporal Knowledge Graph, EventKG).

The team of Professor Liu Ting of Harbin Institute of Technology proposed an abstract event causal network [4] in 2017, using rules to extract causal relationship pairs, and using an ordered set of verbs and nouns to represent events, and finally abstracting each specific event to obtain an abstract event node. Li et al. proposed the concept of Event Evolutionary Graph (EEG)[1] in 2018, extracting narrative event chains from news corpus, and constructing event evolution graphs based on the extracted event chains. The concept of "Event Logic Graph" was first proposed by Professor Liu Ting's team [2]. Han et al. proposed an end-to-end event and event relationship joint extraction model [8] to extract the temporal relationship between events. Causality refers to the occurrence of cause-effect events between two events. In 2017, Choube et al. proposed an iterative method [9] to eliminate the co-referential relationship between events. In 2017, Lu et al. proposed a method [10] to eliminate the co-referential relationships of events based on joint inference learning, which avoids the transfer error between various tasks.

Event recognition and extraction refers to extracting structured events from text. Generally speaking, the basic tasks [3] of event extraction include: event trigger word detection, event type recognition, event element recognition, and event element role recognition. In 2018, Liu et al. proposed a model [11] to jointly extract multiple event trigger words and arguments, which enhances the information flow and can extract multiple events from a sentence. Chen et al. proposed a hierarchical network [12] based on a gated attention mechanism in 2018, and designed the event extraction task as a sequence labeling task. In 2018, Liu et al. used dynamic memory networks [13] to use contextual information for event detection. Yang et al.[14] proposed in 2019 to use a pre-trained language model to extract event trigger words, and use the knowledge obtained from the corpus to generate events.

3. The Proposed Method
3.1. Event Relationship Extraction Template
The event relationship extraction template construction module constructs event related word templates for causality, succession, condition, concurrency, and transition event relationships, so that event pairs can be extracted according to the constructed template. Among them, there are two types of succession relationship extraction templates, ten types of causal relationship extraction templates, one for conditional relationship extraction template, one for transition relationship extraction template and one for concurrent relationship extraction template [15]. For the two kinds of succession relationship extraction templates, the second template has a higher priority than the first template. The sentence matches the second template first. If the match succeeds, the previous and next succession events are obtained. If the match fails, then the first template is matched. The template description is shown in Table 1.

| Succession relationship extraction templates | Related words | Regular expression |
|---------------------------------------------|--------------|--------------------|
| (successional pre-event) (successional related words) (successional post-event) | and then, after, second, then, next | r'(.*)\{0\}\{1\}\{2\}' |

| (successional related words) (successional pre-event) (successional related words) (successional post-event) | [again, again, before, union], [and then]], [[first, first], [second, then]], [[first, first], [again, again, just]], [ [on the one hand], [on the other hand, again, also, still]] | r'\{0\}\{1\}\{2\}' |

3.2. Event Extraction and Structuring Algorithm
The event extraction and structuring algorithm is divided into two parts. First, event extraction algorithm segments each event text in the database into long sentences, completes the subject of the sentence, and performs part-of-speech tagging operation [16], and then relationship extraction template performs event relationship extraction to get event pairs and the related words of the event pair, finally this algorithm segments each event in event pairs to obtain the short sentence set. The time complexity of the algorithm is \( O(mn) \), where \( m \) is the total number of event texts in the database, and \( n \) is the total number of sentences. The following describes the steps of event extraction in detail [17].

1) Part of speech processing
The set of part-of-speech tags to be retained in the short sentence: ["a", "b", "c", "d", "wp", "i", "j", "v", "n", "nh", "ni", "nl", "ns", "nz"], the part-of-speech standard uses the 863 part-of-speech tagging set, and the stop vocabulary includes some meaningless words, such as "oh, and". The negative vocabulary includes some words with negative meaning, such as "shortage, useless". The word filtering algorithm is: For each word in the event sentence, if the word is in the negative vocabulary, or its part of speech is in the part-of-speech tag set and the word does not belong to the stop vocabulary, then keep the word. According to the above-mentioned part-of-speech tag set and two lexicons, the word filtering algorithm is used to filter each word in the short sentence.

2) Rule filters event
For the event sentence after part-of-speech processing, the algorithm first judges whether the sentence length is longer than 2 or equal to 2. If the condition is satisfied, the event judgment is performed, and if it is not satisfied, the empty event is directly returned. The event representation rule stipulates that the event has semantic integrity and conforms to the "subject-predicate structure" or "subject-predicate-object structure". The event representation rules include the following four rules.

3) Thesaurus filters event
Subjectivity refers to: while describing objective facts, people express their feelings, attitudes, and subjective evaluations of facts. Subjective vocabulary includes words such as "feel, hunch, and think". The judgment and filtering rule for using subjective vocabulary is: if an event sentence contains a word in the vocabulary, then the event is not a candidate event and is removed from the set of candidate event sentences.

4) Event sentiment analysis

This paper uses a method based on semantic understanding to analyze the emotional polarity and intensity of the event sentence by using the emotional dictionary [18]. Because we conduct experiments in the financial field, so we use the emotional dictionary in the financial field.

This algorithm calculates the emotional score of each emotional word according to four situations. The first case is degree adverb with negative word and emotional word", the calculation method is shown in formula (1).

$$w = t \times a \times (-1) \times 2$$  \hspace{2cm} (1)

The second case is negative word with degree adverb and emotional word, the calculation method is shown in formula (2).

$$w = t \times 0.5 \times a$$  \hspace{2cm} (2)

The third case is negative word with emotional word, the calculation method is shown as

$$w = t \times (-1)$$  \hspace{2cm} (3)

The fourth case is degree adverb with emotional word, the calculation method is shown as

$$w = t \times a$$  \hspace{2cm} (4)

where $w$ represents the emotional score of each emotional word after calculation, $t$ represents the current score of the emotional word, and $a$ represents the score of the degree adverb. The process of sentiment analysis for event sentences is as follows. first set the sentiment score $w$ of each word in the event sentence to 0, and then judge whether each word is a positive sentiment word, if it is, then increase $w$ by 1, if not, then decrease $w$ by 1.

3.3. Event Similarity Calculation and Fusion Algorithm

We propose an event similarity calculation and fusion algorithm based on edit distance and event vector. According to the event similarity, the events whose similarity meets the set threshold are clustered, and each type of different events with the same meaning is merged. The time complexity of the algorithm is $O(n^2)$, where $n$ is the total number of events in the database. The steps of the algorithm are described in detail below.

1) Read the event set: algorithm read out the events in each pair of event phrase sets in the database and save them to the json file.

2) Events are divided into multiple categories: algorithm put each event into a list to obtain multiple separate categories.

3) Calculate the edit distance: Given two events, algorithm call the function distance. Levenshtein (event1, event2) to calculate the edit distance. The edit distance is the character operand. If the edit distance is less than or equal to the set threshold 2, and the emotional polarity of the two events is the same, then algorithm merge the two categories into one. If the conditions are not met, go to step 4.

4) Calculate the cosine value: algorithm first preprocess the training corpus, and perform word segmentation and remove stop words, and then use TaggedDocument to process each sentence, finally get the processed training corpus. Then we train the Doc2Vec model Error! Reference source not found., and save the trained model. Word segmentation is performed on the event sentence that needs to be represented by a vector, and the vector representation of the event is obtained by loading the model. Finally, algorithm calculate the cosine value $Wv$ of the angle between the two event vectors V1 and V2, as shown in formula (5).
If the cosine value is greater than the set threshold 0.9, then the two classes are merged into one class. If the above condition is not met, the two events are not similar and clustering is not performed.

3.4. Event Trigger Word Extraction and Argument Extraction Model
In this paper, a sequence labeling scheftame based on a pre-training model is used to construct trigger word extraction model and argument extraction model using the BIO labeling method. The former extracts trigger words and recognizes the corresponding event type [11], which is divided into 65 event types, the latter extract the argument and identify the role of the corresponding argument. We use the pre-trained model and Fine-tune API provided by PaddleHub to complete the model construction, training and prediction. The steps of model construction are described in detail below.

4. Experiments and Evaluation

4.1. Dataset
We use the Scrapy framework to write a crawler script, which crawls the travel texts on a travel website, saves them to the MongoDB database, and obtains the test text in the travel field, which is used to construct the Serial Event Logic Graph. In addition, we also crawl the news text of topic events and save them to the MongoDB database. Through the analysis of different news reports on the same topic event, the sequence relationship between the sub-events within the event is sorted out to construct another Serial Event Logic Graph. We downloaded 20,000 financial news texts from the Internet to construct Causal Event Logic Graph, Conditional Event Logic Graph, Transitional Event Logic Graph, and Concurrent Event Logic Graph.

The training data, validation data, and test data used in event trigger word extraction and argument extraction model training come from the Chinese event extraction data set released by Baidu, which contains 65 event types and a total of 17,000 sentences with event information. We extract 12,000 sentences as the training set, 1,500 sentences as the validation set, and 3,500 sentences as the test set.

4.2. Experiment Results
By using the Event Logic Graph construction method, we construct a variety of graphs. The partial subgraphs, node and relationship descriptions of Causal Event Logic Graph, Conditional Event Logic Graph, Transitional Event Logic Graph, and Concurrent Event Logic Graph are generated. In addition, we also construct Serial Event Logic Graph in the travel field, using verb-object phrases as a structured representation of events, and using front-end technologies such as VIS plug-ins to display the graph. Through the analysis of different news reports on the same topic event, we sort out the temporal relationship between the sub-events in the event and construct another Serial Event Logic Graph.

4.3. Evaluation of Experimental Results
The evaluation results of this model and the ERNIE+CRF model on the test set and verification set are explored. The results show that the results of this model on the three evaluation indicators of accuracy, recall, and f1 are better than results of ERNIE+CRF model, so our model can effectively predict the trigger word, type, argument role and other information of the event.
Table 2. Comparison of evaluation results of this model and ERNIE+CRF model

| Model         | Data set     | Precision | Recall | F1     |
|--------------|--------------|-----------|--------|--------|
| ERNIE+CRF    | Test set     | 0.8418    | 0.8779 | 0.8595 |
|              | Validation set| 0.8270    | 0.8680 | 0.8470 |
| Our model    | Test set     | 0.85510   | 0.88624| 0.86660|
|              | Validation set| 0.85004  | 0.88309| 0.86257|

5. Conclusion and Future Work
In this paper, we propose an Event Logic Graph construction method for event profile. By constructing event relationship extraction templates, using the proposed event extraction and structuring algorithms, event similarity calculation and fusion algorithms, and event trigger word extraction and argument extraction models, we finally construct Serial Event Logic Graph, Causal Event Logic Graph, Conditional Event Logic Graph, Transitional Event Logic Graph, and Concurrent Event Logic Graph. Event Logic Graph [6] describes the logical society, and the research object is the logical relationship between predicate event phrases and events.

References
[1] Li Z., Ding X., Liu T. Constructing Narrative Event Evolutionary Graph for Script Event Prediction. International Joint Conference on Artificial Intelligence, 2018: 4201-4207.
[2] Ding X., Li Z., Liu T. ELG: An Event Logic Graph[J]. arXiv: Artificial Intelligence, 2019.
[3] Wei X. Overview of the Construction Technology and Application of Event Knowledge Graph. Computer and Modernization, 2020(01): 10-16.
[4] Zhao S., Wang Q., Massung S. Constructing and Embedding Abstract Event Causality Networks from Text Snippets. Proceedings of Conference on Web Search and Data Mining, 2017: 335-344.
[5] Goran G., Jan S. Construction and evaluation of event graphs. Natural Language Engineering, 2014, 21(04). 1-46.
[6] Rospocher M, Van Erp M, Vossen P, et al. Building Event-Centric Knowledge Graphs from News. Ssmr Electronic Journal, 2016: 37-38.
[7] Gottschalk S., Demidova E. EventKG - the Hub of Event Knowledge on the Web - and Biographical Timeline Generation. Social Work, 2019, 10(06). 1039-1070.
[8] Han R., Ning Q., Peng N. Joint Event and Temporal Relation Extraction with Shared Representations and Structured Prediction. arXiv: Computation and Language, 2019.
[9] Choubye P K, Huang R. TAMU at KBP 2017: Event nugget detection and coreference resolution. arXiv preprint arXiv:1711.02162, 2017.
[10] Lu J, Ng V. Joint Learning for Event Coreference Resolution. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2017. 90-101.
[11] Liu X., Luo Z., Huang H. Jointly multiple events extractions via attention-based graph information aggregation. arXiv preprint arXiv:1809.09078, 2018.
[12] Chen Y., Yang H., Liu K. Collective Event Detection via a Hierarchical and Bias Tagging Networks with Gated Multi-level Attention Mechanisms. Proceedings of Conference on Empirical Methods in Natural Language Processing, 2018. 1267-1276.
[13] Liu S., Cheng R., Yu X. Exploiting Contextual Information via Dynamic Memory Network for Event Detection. Proceedings of Conference on Empirical Methods in Natural Language Processing, 2018. 1030-1035.
[14] Yang S., Feng D, Qiao L. Exploring Pre-trained Language Models for Event Extraction and Generation. Proceedings of Annual Meeting of the Association for Computational Linguistics, 2019: 5284-5294.
[15] Gao L., Zhou G., Luo J. A review of meta-event extraction research. Computer Science, 2019, 46(08): 9-15.

[16] Guo Z., Zhang Y., Chen U. Character-Level Dependency Model for Joint Word Segmentation, POS Tagging, and Dependency Parsing in Chinese. IEICE Transactions on Information and Systems, 2016, 99(01): 257-264.

[17] Li F. Overview of Chinese Word Segmentation Technology. Modern Computer (Professional Edition), 2018, 634(34): 19-22.

[18] Wu J., Lu K. Chinese Weibo sentiment analysis based on multiple sentiment dictionaries and rule sets. Computer Applications and Software, 2019, 36(09): 93-99.

[19] Le V., Mikolov T. Distributed Representations of Sentences and Documents. arXiv: Computation and Language, 2014.