

Research article

Biomedical event argument detection method based on multi-feature fusion and question-answer paradigm

Jinghan Tian, Shuai Xing, Qianmin Su *

School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, PR China

ARTICLE INFO

Keywords:

Biomedical event extraction

Multi-feature

Question-answering paradigm

ABSTRACT

Objective: To address the challenges arising from the rapid growth of text data in the biomedical field, including the problems of irrelevant argument interference and deep semantic association neglect in existing event argument detection methods, as well as the difficulty of multiple event extraction. We aim to propose a new event argument detection method that can accurately mine valuable information from biomedical texts through multi-feature fusion and the question-and-answer paradigm, while also addressing the limitations of existing methods.

Methods: We propose an event argument detection method based on multi-feature fusion and the question-answer paradigm. First, we split each event in the sentence into an independent question-and-answer format to reduce the complexity of detection. Then, in order to reduce the interference of irrelevant arguments, we use syntactic distance and external prior knowledge to find the corresponding argument prior template for each event type. Next, we introduce the multi-feature attention mechanism to fully explore the deep semantic features. Finally, we formulate post-processing methods for predefined event structures to form final biomedical events.

Results: On the MLEE dataset, our model achieved 62.50% in event extraction of F1 scores, which is superior to other advanced event extraction methods.

Conclusion: Our method achieves good performance in the event extraction task and provides strong support for the mining of valuable information in biomedical texts.

Statement of significance

With the wide application of deep learning, many neural network models have been used for biomedical event argument detection and are significantly superior to traditional feature-based models. However, most of the biomedical event argument tests still have different defects. This paper proposes a biomedical event argument detection method based on multi-feature fusion and question-and-answer paradigm, and designs post-processing rules according to predefined event structures to generate structured event representations. Experimental results also verify the effectiveness of the method presented in this chapter.

1. Introduction

In the field of biomedicine, biomedical literature is growing at an alarming rate of more than one million per year, which provides huge data resources for medical researchers. However, how to extract discrete information closely related to current research from

* Corresponding author at: University of Engineering Science, No. 333 Longteng Road, Shanghai, China.

E-mail addresses: 18333838103@163.com (J. Tian), xingshuai68@126.com (S. Xing), suqm@sues.edu.cn (Q. Su).

<https://doi.org/10.1016/j.heliyon.2024.e34057>

Received 16 January 2024; Received in revised form 23 May 2024; Accepted 3 July 2024

Available online 21 July 2024

2405-8440/© 2024 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).



Fig. 1. Contains an example of biomedical text for an event.

these massive medical corpora has become a hot and difficult issue. Therefore, biomedical event extraction technology has emerged, aiming at extracting multi-semantic relations between fine-grained biological entities and presenting detailed information of events in a structured way, which is of great value to drug development and disease prevention. Biomedical event extraction technology can be used to construct biological pathway networks [1,2]. And quickly locate drug information [3,4]. It can also be used for the construction of medical knowledge maps to increase the breadth, depth, practicality and accuracy of the map [5].

Biomedical events generally consist of an event type, a trigger word, arguments, and argument roles. A trigger word is typically a verb or phrase that signifies the occurrence of a specific event. Arguments are usually biomedical entities or other trigger words (nested events), and argument roles describe the relationship between the arguments and their participation in the event. For example, the sentence in Fig. 1 contains two events: 1) Event Type: Regulation, Trigger Word: altered, Theme Argument: bFGF, Cause Argument: treatment. 2) Event Type: Regulation, Trigger Word: altered, Theme Argument: bFGF, Cause Argument: treatment. Biomedical event extraction can be divided into two steps: trigger word detection and argument detection. Biomedical event argument detection is the second step of event extraction, aiming to identify the role associations between trigger words and event arguments. After completing argument detection, structured biomedical events are constructed based on specific post-processing rules. The results of event argument detection directly influence the final event composition.

Currently, existing methods for biomedical event argument detection primarily include rule-based methods, machine learning-based methods, and deep learning-based methods.

(1) Rule-based and Machine Learning-based Methods

Rule-based methods offer high accuracy and interpretability, but developing and maintaining the rules require substantial human labor and time. Additionally, these methods are not well-suited for handling complex semantic structures and have limited generalization capabilities. To address this, Pyysalo et al. [6] used an SVM classifier as a baseline model, achieving biomedical event extraction by incorporating a rich set of manually designed features. Zhou et al. [7] designed a semi-supervised learning model based on latent topics, which automatically generates event annotations from unannotated biomedical data by leveraging hidden topic information. In contrast, machine learning-based methods provide higher accuracy and generalization capabilities but are unable to handle complex semantics and require domain-specific knowledge for feature engineering.

(2) Deep Learning-based Methods

Björne et al. [8] developed an argument detection framework utilizing convolutional neural networks (CNN) and dependency parsing. Yan et al. [9] proposed a bottom-up event extraction framework. He et al. [10] exploited various levels of semantic features through attention mechanisms between words and sentence-level attention mechanisms related to event arguments. Zhu et al. [11] introduced an extraction framework based on a hybrid neural network ensemble strategy. Li et al. [12] incorporated dependency-based word embeddings and pre-trained word embeddings. Li et al. [13] introduced a novel Tree-LSTM (Long Short-Term Memory) network driven by an external knowledge base. Zhao et al. [14] proposed a document-level biomedical event extraction framework. Huang et al. [15] proposed a Graph Edge Conditioned Attention Network (GEANet) that organizes knowledge into a hierarchical knowledge graph representation. Li et al. [16] presented a Gated Recurrent Unit (GRU) network framework for domain-specific word representation. Wang et al. [17] employed a multi-round question-answering approach for argument detection. Shen et al. introduced a multi-head attention mechanism to capture semantic relationships between words. Additionally, Zhao et al. [18] introduced a novel Reinforcement Learning (RL) framework, marking the first attempt to solve biomedical event extraction tasks using reinforcement learning. Shen et al. [19] proposed a joint event extraction method based on CNN-BiGRU and attention mechanisms. Gong et al. [20] proposed a joint biomedical event extraction model named BGHGCN (BiGraph Hybrid Graph Convolutional Network).

However, existing detection methods treat all candidate arguments equally, disregarding the prior information between event types and candidate arguments. This oversight results in the generation of numerous negative samples during training, which hampers the model's ability to learn more fundamental features. Secondly, current methods overlook the fine-grained semantic interactions between local and global perspectives, directly affecting the performance of model predictions. Lastly, most methods typically process multiple events within a single sentence uniformly, increasing detection difficulty and neglecting the substantial benefits that additional contextual information could provide for extraction.

To address the aforementioned issues, we propose a biomedical event argument detection method based on multi-feature fusion and the question-answering paradigm. Firstly, we decompose each event in the sentence into independent question-answer forms and tailor corresponding question templates for each event. Secondly, we utilize syntactic distance and external prior knowledge to identify prior argument templates for each event type, preserving candidate arguments that meet the criteria and reducing interference from irrelevant arguments. Then, we introduce a multi-feature attention mechanism to fully explore the potential semantic representations between trigger words and candidate arguments, trigger word-candidate argument pairs, and trigger word-candidate argument pairs with roles, thereby preventing critical semantic loss. Finally, we combine multiple features to answer the arguments and their role information corresponding to each event. Results on the MLEE corpus demonstrate that the candidate argument generation method,

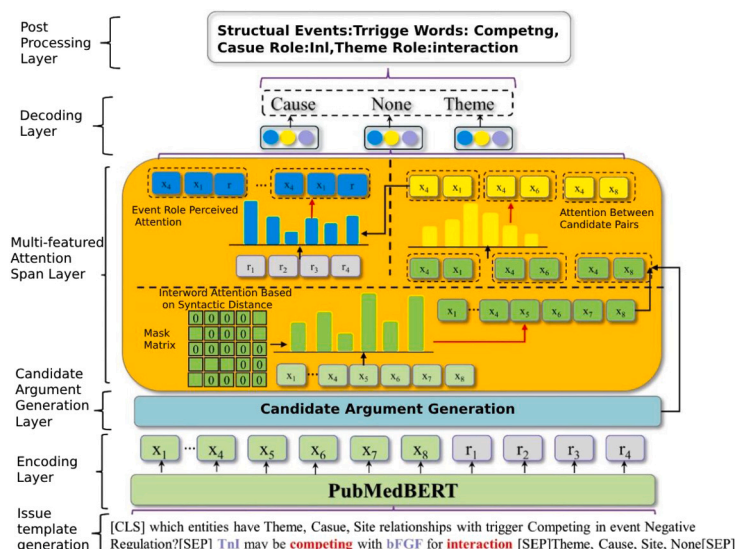


Fig. 2. Contains an example of biomedical text for an event.

multi-feature attention, machine reading comprehension paradigm, and proposed post-processing rules effectively improve model performance. The proposed model achieves an F1-score of 62.50%, outperforming other baseline models.

In summary, our contributions in this paper are in three folds:

- We propose a machine reading comprehension approach to decompose each event in the sentence into independent question-answer forms, reducing detection complexity while resolving ambiguity between events.
- The argument prior templates generated through syntactic distance and external prior knowledge have the capability to mitigate interference from irrelevant arguments, thereby enhancing the model's generalization ability and robustness.
- Leveraging a multi-feature attention mechanism allows for the comprehensive exploration of potential semantic associations between words, candidate pairs, and candidate pairs with roles, thus mitigating the risk of critical semantic loss.

2. Method

2.1. Detection method

As illustrated in Fig. 2, the proposed biomedical event argument detection method consists of six main components. In the Question Template Generation Layer, simple question templates are generated for each specified event type based on predefined argument roles. In the Encoding Layer, the question templates, original text, and corresponding argument role prototype lists are concatenated to form the overall query vector representation. In the Candidate Argument Generation Layer, candidate arguments are selected for each event type from all entities and trigger words in the text using syntactic distance and prior knowledge of event-argument relationships in the training set. Irrelevant arguments are filtered out to ensure relevance to the current event. In the Multi-feature Attention Layer, potential semantic features between trigger words and candidate arguments, candidate pairs (trigger word-candidate argument pairs), and candidate pairs with roles are explored to achieve fine-grained interaction at both local and global levels, thereby extracting richer semantic representations in biomedical events. In the Decoding Layer, multiple learned feature representations are concatenated and used to predict the role categories of candidate pairs through a softmax classifier. In the Event Post-processing Layer, valid combinations of event trigger words and candidate arguments that meet post-processing rules are selected as the final biomedical events.

2.1.1. Generation of question templates

In the current study, biomedical event argument detection is framed as a classification-based relation extraction task, aiming to extract effective relationships between trigger words and arguments. However, this classification approach requires ample data support to ensure model performance and overlooks the substantial assistance that additional information could provide for model predictions. Moreover, when multiple events exist within a single sentence, this method may lead to ambiguity issues for the model. The machine reading comprehension-based question-answering learning paradigm adopted in this paper effectively addresses the aforementioned challenges. The essence of this method lies in generating independent question templates for each event in the sentence and interacting them with the original text to answer specific relationships between two words in the text. This approach not only introduces question template information to enhance the reasoning process within the context but also enriches the training set with effective combinations of a medical text and multiple question templates, thereby alleviating data scarcity issues and detection complexity.

Table 1
Templates for questions on 19 event types.

Event type	Predefined roles	Question template
Cell proliferation	Theme	Which entities have Theme relationships with trigger {} in event cell promotion?
Development	Theme	Which entities have Theme relationships with trigger {} in event development?
Blood Vessel Development	Theme, At-Loc	Which entities have Theme, At Loc relationships with trigger {} in event Blood Vessel Development?
Death	Theme	Which entities have Theme relationships with trigger {} in event Death?
Breakdown	Theme	Which entities have Theme relationships with trigger {} in event breakdown?
Remodeling	Theme	Which entities have Theme relationships with trigger {} in event removal?
Growth	Theme	Which entities have Theme relationships with trigger {} in event growth?
Synthesis	Theme	Which entities have Theme relationships with trigger {} in event synthesis?
Gene Expression	Theme	Which entities have Theme relationships with trigger {} in event Gene Expression?
Transcription	Theme	Which entities have Theme relationships with trigger {} in event Transcription?
Catabolism	Theme	Which entities have Theme relationships with trigger {} in event Catabolism?
Phosphorylation	Theme, Site	Which entities have Theme, Site relationships with trigger {} in event Photosynthesis?
Dephosphorylation	Theme, Site	Which entities have Theme, Site relationships with trigger {} in event Dephosphylation?
Localization	Theme, At/From/To-Loc	Which entities have Theme, At/From/To Loc relationships with trigger {} in event localization?
Binding	Theme, Site	Which entities have Theme, Site relationships with trigger {} in event binding?
Regulation	Theme, Cause, Site	Which entities have Theme, Cause, Site relationships with trigger {} in event regulation?
Positive Regulation	Theme, Cause, Site	Which entities have Theme, Cause, Site relationships with trigger {} in event Positive Regulation?
Negative Regulation	Theme, Cause, Site	Which entities have Theme, Cause, Site relationships with trigger {} in event Negative Regulation?
Planned Process	Theme, Instrument	Which entities have Theme, Instrument relationships with trigger {} in event Planned Process?

One of the key challenges in machine reading comprehension is how to generate effective question templates. In argument detection, most machine reading comprehension approaches typically use simple manually designed templates, where questions are always framed as “What is the relationship between a certain trigger word and a certain argument?” or “Who is the argument related to a certain trigger word?” Such templates are often tailored for classifying relationships between individual trigger words and arguments, overlooking the potential correlations between candidate pairs within the same event. To address this limitation, this paper generates question templates based on predefined argument roles for each event type: “which entities have {} relationships with trigger {} in event {}?”, where the three {} placeholders are filled from left to right with the predefined argument role corresponding to the current event type, the trigger word corresponding to the current event, and the name of the current event type, respectively. This question template not only standardizes the role, trigger word, and event type information needed for classification but also represents a complete event question query. Therefore, it facilitates feature sharing among multiple candidate pairs within the same event and benefits the model’s reasoning process. Table 1 presents the question templates for different event types in the biomedical event extraction corpus MLEE.

2.1.2. Context coding

Given a biomedical document consisting of L sentences, each sentence containing n words $W = \{w_1, w_2, \dots, w_n\}$, with the assumption that there are m trigger words present in the sentences. For a trigger word w_i , we select the corresponding question template t_i from Table 1, and fill it into the template to form the final query $Ques_i$. The event type t_i corresponds to a predefined list of role names and the non-role type name “None”, which forms the event role prototype list $R_i = \{role^1_i, \dots, role^k_i\}$ used for argument detection. In this paper, the question template $Ques_i$ for argument detection, the original text W , and the event role prototype list R_i are concatenated into a sequential sequence R_i , as shown below:

$$Input A_i = [CLS] Ques_i [SEP] Sentence [SEP] R_i [SEP] \quad (1)$$

In this context, [CLS] is the first token of the input sequence, used to represent the aggregate representation of the sentence, and [SEP] is the separator token between different sentences, used to concatenate two sentences into a single input sequence. We utilize the pre-trained PubMedBERT to encode the aforementioned sequence in order to obtain the contextual representation $X = \{x_1, x_2, \dots, x_n\}$ of each word in the input text and the contextual representation $R = \{r_1, r_2, \dots, r_n\}$ of each role type in the event role prototype list. Note that PubMedBERT employs the WordPiece tokenizer, a subword segmentation method that splits input words into a series of subword units. Therefore, we select the first subword segment of a split word as the contextual representation of that word. The specific formula is as follows:

$$\{x_1, x_2, \dots, x_n, r^1_i, r^2_i, \dots, r^{k+1}_i\} = \text{PubMedBERT}(\text{Input } A_i) \quad (2)$$

2.1.3. Candidate argument generation layer

In biomedical documents, a sentence may contain multiple trigger words and multiple entities simultaneously. Additionally, due to the specificity of biomedical events, in simple events, all event arguments are composed of biomedical entities, while in complex events, there are other trigger words in the event arguments. Since this paper views event argument detection as a relation extraction task based on machine reading comprehension, it is necessary to establish trigger word-entity pairs and trigger word-trigger word pairs as prerequisites for fine-grained relation extraction tasks. Simply enumerating all valid combinations of trigger words and entities in a sentence would generate a large number of negative samples. Furthermore, under different sentences, the relationship

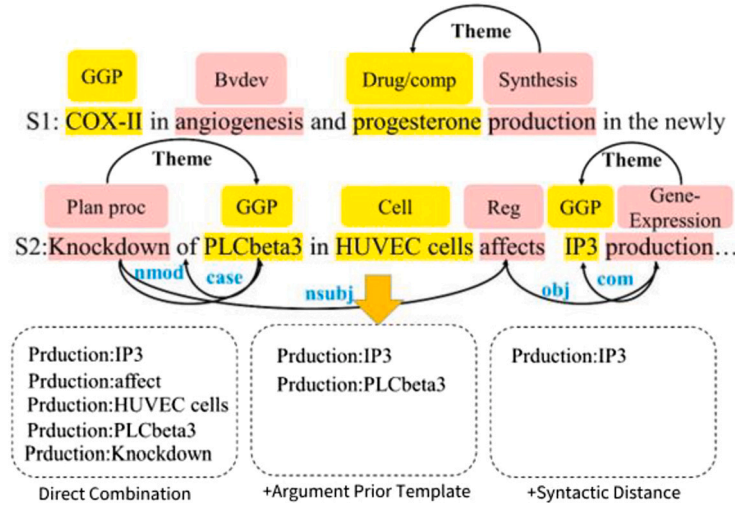


Fig. 3. Contains an example of biomedical text for an event.

between the same trigger word and the same entity type may be inconsistent, leading to ambiguous model predictions and affecting its generalization ability. As shown in Fig. 3, in sentence S1, there is no role relationship between the event trigger word “production” and the gene or gene product (GGP) entity “COX-II”, however, in sentence S2, there is a Theme role relationship between the event trigger word “production” and the gene or gene product entity “IP3”. To address the above issues, this paper proposes a Candidate Argument Generation Layer, which mainly consists of three steps. First, learn the argument prior templates corresponding to specific event types from the training documents. Then, filter out the arguments from the sentence that do not conform to the argument prior templates. Finally, further filter out arguments that do not meet the dependency constraints based on the dependency relationships of the sentence, leaving the remaining arguments as candidate arguments for the event. Assuming the current sentence is S , the event type in the event question template is t_i , and the event trigger word is w_i , the specific method is as follows:

In the first step, to obtain the argument prior template T_i for event type t_i , each entity and trigger word in the sentence is labeled with the corresponding entity and event type names, while non-entities and non-trigger words are labeled with O . Then, all events in the training document are traversed, and if the event type of the current event is t_i , all entity type labels/event type labels corresponding to the entities/trigger words under the event are added to T_i . After the traversal, duplicate elements in T_i are removed, and the deduplicated T_i is used as the argument prior template for event type t_i .

In the second step, to filter out event argument elements not related to the current event type, all entities and trigger words in the sentence S except the trigger word w_i are first counted and added to the list EAT_i . Then, iterate through all elements in EAT_i and remove the current entity/trigger word from EAT_i if its corresponding entity type label/event type label does not exist in T_i .

In the third step, to obtain the final candidate thesis element under the event, firstly, the syntactic distance matrix $SDM_i \in \mathbb{R}^{n \times r}$ between words is calculated based on the dependency relationship of sentence S , where SDM_{ij} denotes the syntactic distance from word w_j to word w_i . Then all the elements in the EAT_i are traversed. If the syntactic distance between the current element and the event trigger word w_i is greater than the syntactic distance threshold β of the candidate thesis element, then they cannot form a candidate pair, and therefore the element is removed from the EAT_i . After the traversal, the remaining elements in EAT_i are the candidate arguments to be detected under the current event.

Table 2 presents the prior templates of event types, where “+” represents the remaining unlisted types. It can be observed that a specific type of event only allows specific types of entities or trigger words as event arguments. For instance, the event type “Cell proliferation” only allows entities of type “Cell” as arguments. Additionally, only four event types, namely “Regulation”, “Negative Regulation”, “Positive Regulation”, and “Planned Process”, allow other events as arguments. Therefore, the learned prior templates of arguments from training documents not only filter out some entities or trigger words that do not meet the criteria but also distinguish between simple events and complex events based on global information about event types and structures. However, such prior templates are based on global information about event types and structures, neglecting the local information in different contexts.

Referring to the dependency parse of sentence “S2” in Fig. 3, a dependency edge can be observed between the trigger word “production” and the argument “IP3”. The syntactic distance between “production” and “PLCbeta3” is 3. Although “PLCbeta3” is not an argument of “production”, it and “IP3” are entities of the same type, both included in the prior template of arguments for the event type “Gene-Expression”. Therefore, the introduction of syntactic distance can uncover potential information at the sentence level and further filter out irrelevant candidate pairs.

As illustrated in Fig. 3, sentence S2 contains 3 trigger words and 3 entities. For the trigger word “production”, which triggers the event type “Gene-Expression”, according to the prior templates of argument types in step 1, it can only form candidate pairs with “IP3” and “PLCbeta3”. Referring to the syntactic distance for further filtering, assuming a given syntactic distance threshold of 2, the syntactic distance between “production” and “PLCbeta3” is greater than 2, indicating that they cannot form a candidate pair. Thus,

Table 2
Argument apriori templates for 19 event types.

Events	Argument apriori templates
Cell proliferation	Cell
Development	Cell, Cellular_component, Tissue, Organ, Multi-tissue_structure, +
Blood Vessel Development	Tissue, Multi-tissue_structure, Organ, Anatomical_system, +
Death	Cell, Pathological_formation, Tissue, Organism
Breakdown	Cell, Cellular_component, Tissue, Organ, +
Remodeling	Cell, Cellular_component, Multi-tissue_structure
Growth	Cell, Pathological_formation, Tissue, Multi-tissue_structure
Synthesis	Drug_or_compound
Gene Expression	Gene_or_gene_product
Transcription	Gene_or_gene_product
Catabolism	Gene_or_gene_product, Drug_or_compound
Phosphorylation	Gene_or_gene_product
Dephosphorylation	Gene_or_gene_product
Localization	Cell, Tissue, Organ, Drug_or_compound, +
Binding	Cell, Cellular_component, Tissue, Gene_or_gene_product, +
Regulation	Negative_regulation, Dephosphorylation, Binding, Breakdown, +
Positive Regulation	Blood_vessel_development, Gene_expression, Death, Binding, +
Negative Regulation	Blood_vessel_development, Growth, Synthesis, Breakdown, +
Planned Process	Blood_vessel_development, Organ, Cell, Tissue, +

the sentence generates only 1 candidate pair. If we directly combine trigger word-entity pairs and trigger word-trigger word pairs in the sentence, this event would produce 5 candidate pairs. If syntactic distance is not considered, it would result in 2 candidate pairs.

2.1.4. Multi-feature attention layer

To better leverage contextual features crucial in sentence understanding, consider semantic relevance among candidate pairs and between candidate pairs and argument roles, and prevent local or global semantic loss, this paper proposes a multi-feature attention mechanism. This mechanism captures rich semantic representations from various angles, including the following three features: 1) Contextual semantic features between words: Semantic information in the sentence is extracted by utilizing contextual keywords to discern the specific meaning of the trigger word or candidate argument in the altered context. 2) Associative semantic features between candidate pairs: The semantic relationship between candidate pairs is identified, and their correlation is magnified by computing their similarity. 3) Role-aware semantic features between candidate pairs and argument roles: Argument role information is employed to assess whether the candidate pair is a suitable argument, and their semantic interaction is further elaborated from a relational perspective.

(1) Contextual feature representation between words

In evolving contexts, trigger words or candidate arguments may convey varying meanings, triggering different events or assuming different roles. Consequently, the detection of event argument elements relies on context-specific information interactions. However, the contextual features integrated by simple attention mechanisms often contain abundant redundant information unrelated to events, potentially diluting the significance of keywords in candidate argument features and trigger word features. To prioritize crucial event-related features, when the syntactic distance between two words in a sentence exceeds the inter-word syntactic distance threshold γ , they become unable to engage in semantic information interaction. Leveraging the syntactic distance matrix SDM , this approach constructs a masking matrix $MASK$:

$$MASK_{ij} = \begin{cases} 0, & SDM_{ij} \leq \alpha \\ -\infty, & \text{otherwise} \end{cases} \quad (3)$$

$$\text{score}(x_i, x_j) = \frac{1}{\sqrt{d_j}} \tanh(x_i \cdot w_w) \cdot x_j \quad (4)$$

$$e_{ij} = \text{softmax}(\text{score}(x_i, x_j) + MASK_{ij}) \quad (5)$$

$$C_i^w = \sum_j^n e_{ij} \cdot x_j \quad (6)$$

where $W_w \in R^{d_j \times d_j}$ is the parameter matrix, d_j is the embedding dimension of x_j , $\text{score}(x_i, x_j)$ denotes the attention score between the word vectors x_i and x_j , e_{ij} denotes the normalized attention weights between x_i and x_j , and x_i^w is the contextual feature representation of the word w_i .

(2) Associative semantic feature representation between candidate pairs

Under the same event, there is usually a strong semantic correlation between candidate pairs, and mining this correlation information can help the model to recognize the correct candidate thesis elements to construct the complete event structure. After the inter-word contextual attention mechanism, the word vector features in a sentence are represented as $C^w = \{C_1^w, C_2^w, \dots, C_n^w\}$, and their corresponding contextual vector representations can be obtained based on the positional information of the event trigger words

and candidate theses. Since trigger words and candidate arguments may span multiple tokens in the sentence, one or more vectors corresponding to trigger words w_i and candidate arguments w_j are maximally pooled to obtain the final trigger word feature vector C_i^t and candidate argument feature vector C_j^a . In order to reasonably fuse the vector representations between specific trigger words and candidate arguments in a sentence to obtain candidate pairs, this paper performs nonlinear activation on C_i^t and C_j^a respectively, and the activated vectors are subjected to a splicing operation in order to obtain by a candidate pair feature vector C_{ij}^p , which is formulated as follows:

$$C_i^t = \text{MaxPool} (C_i^w, C_{i+1}^w, \dots, C_{i+l}^w) \quad (7)$$

$$C_j^a = \text{MaxPool} (C_j^w, C_{j+1}^w, \dots, C_{i+m}^w) \quad (8)$$

$$C_{ij}^p = \left[\tanh (C_i^t \cdot w_i), \tanh (C_j^a \cdot w_a) \right] \quad (9)$$

Where $\text{MaxPool}(\cdot)$ is the maximum pooling function, l and m are the number of tokens occupied by w_i and w_j respectively, and w_t and w_a are the parameter matrices. For the fused candidate pair feature vector list $C^p = \{C_{i1}^p, C_{i2}^p, \dots, C_{iz}^p\}$, this paper further utilizes the attention mechanism between candidate pairs to obtain the associative semantic feature representation between each candidate pair and other candidate pairs, as shown below:

$$\text{score} (C_{ij}^p, C_{ik}^p) = \frac{1}{\sqrt{d_i}} \tanh (C_{ij}^p \cdot w_p) \cdot C_{ik}^p \quad (10)$$

$$\alpha_{jk}^p = \text{softmax} (\text{score} (C_{ij}^p, C_{ik}^p)) \quad (11)$$

$$H_{ij}^p = \sum_k \alpha_{jk}^p \cdot C_{ik}^p \quad (12)$$

where w_p is the parameter matrix, d_i is the embedding dimension of C_{ij}^p , $\text{score}(C_{ij}^p, C_{ik}^p)$ denotes the attention score between the candidate pair of feature vectors C_{ij}^p and C_{ik}^p , α_{jk}^p denotes the normalized attentional weights between C_{ij}^p and C_{ik}^p , and H_{ij}^p is the associative semantic feature vector of C_{ij}^p .

(3) Role-aware semantic features between candidate pairs and roles

Previous studies have shown that the association information between argument roles and entities is crucial for argument extraction. In this paper, this association is extended from the entity level to the candidate pair level. To capture the potential association representation between candidate pairs and argument roles, this paper uses the event role prototype list as an auxiliary decision, effectively utilizes the predefined role information of each event type to limit the interference of irrelevant roles, and applies the attention mechanism to learn the weight distribution of each candidate pair on the event role prototype list to generate role-aware semantic features for each candidate pair:

$$\text{score} (H_{ij}^p, r_i^k) = \frac{1}{\sqrt{d_k}} \tanh (H_{ij}^p \cdot w_{pr}) \cdot r_i^k \quad (13)$$

$$\alpha_{jk}^{pr} = \text{softmax} (\text{score} (H_{ij}^p, r_i^k)) \quad (14)$$

$$H_{ij}^r = \sum_k \alpha_{jk}^{pr} \cdot r_i^k \quad (15)$$

of which w_{pr} is the parameter matrix, d_k is the embedding dimension of r_i^k , $\text{score}(H_{ij}^p, r_i^k)$ denotes the attention score between the candidate pairwise associative semantic feature vector H_{ij}^p and the role-specific vector r_i^k , α_{jk}^{pr} denotes the normalized attentional weights between H_{ij}^p and r_i^k , and H_{ij}^r is the role-aware semantic feature representation of C_{ij}^p .

2.1.5. Argument detection decoding layer

After completing the multi-feature attention layer, it is necessary to detect the candidate pairs, that is, to determine the argument role corresponding to each candidate pair. H_{ij}^p The event type vector corresponding to this event encoded by one-hot ET_i and the role-perception feature vector of the candidate pair H_{ij}^r Softmax activates the fully connected network of functions to locate the role class probability that each candidate pair plays in a given event:

$$p(y_i | x_i, x_j) = \text{softmax} (W_{ta} [H_{ij}^p, ET_i, H_{ij}^r] + b_{ta}) \quad (16)$$

among them, W_{ta} is the transfer parameter matrix, b_{ta} is the bias vector, and $p(y_i | x_i, x_j)$ expresses the word w_i and word w_j 's role type y_i class probability.

2.2. Post-treatment

After detecting the event arguments corresponding to each event in the sentence, it is necessary to effectively combine these recognized arguments with trigger words to form a complete biomedical event. The purpose of the post-processing layer is to identify

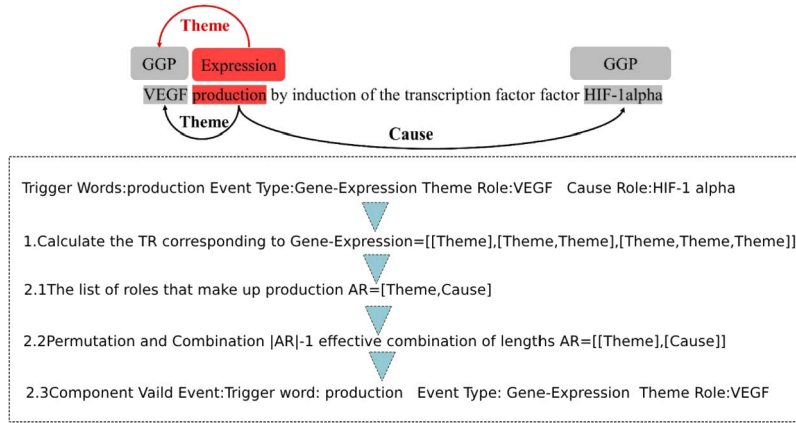


Fig. 4. Post-processing process instance.

Table 3
Statistical information for the MLEE dataset.

Label	Training set	Validation set	Test set	All of it
Documentation	131	44	87	262
sentence	1271	457	880	2608
event	3296	1175	2206	6677

events that meet specific structural definitions. Although this paper utilizes machine reading comprehension to obtain all argument information for an event, there may still be invalid relations within the argument detection results, which can lead to the construction of invalid events. To address this issue, the argument role composition structure of each event type in the training document is analyzed, and corresponding post-processing rules are designed to filter out candidate events that do not meet the criteria, thereby generating the final structured event. Suppose the current event type to be constructed is t . The post-processing method employed in this paper involves the following steps:

First, in order to obtain the argument role structure template corresponding to the event type, we need to traverse all the events in the training document. t TR If the event type of the current event is, the argument role structure for that event is added to. t TR Once the traversal is complete, all of the structures in, form the argument role structure template for the event type.

In the second step, the role list AR is composed of the role information of all the identified thesis elements in the current event. If AR already exists in the TR , then the current event trigger word and the recognized theses can be directly composed into a structured event. If the role list AR is not in the TR , but is also not empty, the information in AR needs to be permuted and combined to arrive at a valid combination \overline{AR} of all elements of length $|AR|-1$. Iterate through all the combinations in \overline{AR} , and if the current combination is in TR , compose an event with all the corresponding arguments and trigger words under the current combination. If the current combination is not in TR , the combination is discarded, i.e., the current event is considered as a non-valid combination.

Fig. 4 shows an example flow of post-processing, taking the sentence “VEGF production by induction of the transcription factor HIF-1alpha” as an example, in which production is the trigger word for the type of Gene Expression event. In this sentence, production is the trigger word for the Gene Expression event type. The event theses Theme: VEGF and Cause: HIF-1alpha have been identified in the detection. In the post-processing stage, this paper first counts the role structure templates $TR = [[Theme], [Theme, Theme], [Theme, Theme, Theme]]$ corresponding to Gene Expression event types. The role information for all theses is then computed to form the role list $AR = [Theme, Cause]$. Since AR is not in the TR , a backward compatibility operation is required. Arrange all roles in AR and get all valid combinations of length $|AR|-1$ from them $\overline{AR} = [[Theme], [Cause]]$. In \overline{AR} , since $[Theme]$ exists in TR , the following valid events are generated: trigger word: production, event type: Gene-Expression, Theme role: VEGF.

3. Experiment

3.1. Experimental setup

3.1.1. Dataset

In order to test the performance of our model in biomedical event extraction, we implement the model in MLEE (Multi-Level Event Extraction) [6]. The MLEE corpus was proposed by Pyysalo in 2012 and is one of the most widely used event extraction corpora in the biomedical field. MLEE contains 262 biomedical documents, 2608 sentences and 6677 multi-level biomedical events. In addition, it also provides a data set partition standard, that is, the training set, the validation set and the test set are divided according to 3:1:2 ratio division, the specific data division is shown in Table 3.

3.1.2. Parameter setting

The model is implemented based on pre-trained PubMedBERT, where batch_size is set to 16, epoch is set to 20, learning rate is set to 5 e-5, and candidate argument syntactic distance threshold β is positioned at 3, and the inter-word syntactic distance γ is set to 4. Stanford parser [21] is used to generate the dependency tree structure for each sentence in the MLEE, using Adam stochastic gradient descent algorithm, the maximum length of each batch determines the maximum sentence length.

3.1.3. Evaluation metrics and baselines

In experiments, three metrics, Precision (P), Recall (R) and F1-score (F1), are utilized to evaluate the performance of biomedical event extraction. In order to verify the performance of the proposed biomedical event extraction model, it is compared with the following representative methods:

(1) Pyysalo et al. [6] using SVM classifier as baseline model, biomedical event extraction is realized by combining rich artificial design features.

(2) Zhou et al. [7] this paper proposes a semi-supervised learning model based on hidden topic, which can automatically generate event annotations from unannotated biomedical data by using hidden topic information, and improve the performance of biomedical event extraction by expanding training samples.

(3) Björne et al. [8] the word embedding, POS label embedding and dependency path embedding are integrated into the convolutional neural network to perform the relation classification task, and the whole model is integrated into the TEES open source framework to extract biomedical events and relations.

(4) Li et al. [13] in order to make better use of the context encoding information, a gating mechanism is used to automatically incorporate the ontology features of the external knowledge base into the hidden state of Tree-LSTM, and the shortest dependency path is used to find the dependency relationship between the current given trigger word and all candidate arguments, so as to assign the argument role to each candidate pair.

(5) Yan et al. [9] without the need of complex dependencies, the annotated event argument information is used to train the context embedding, and the lexical information on both sides of the event parameter is trained in a bottom-up manner to complete the complete event extraction.

(6) Zhu et al. [11] the character embeddings are learned by CNN, and event-triggering words and relations are detected from the original text in an end-to-end manner by BiLSTM. Finally, events are constructed by reducing error accumulation based on the proposed combination strategy for solving the optimization problem.

(7) He et al. [10] the proposed multi-level attention mechanism extends from word-level attention to sentence-level attention, which effectively utilizes the correlation between arguments to improve the extraction performance.

(8) Zhao et al. [14] a document-level biomedical event extraction framework is proposed. Dependency based GCN is used to capture local context features. Hypergraph is used to model global context features. Fine-grained interaction between local and global contexts is realized by stacking hypergraph aggregation neural network modules.

(9) Li et al. [12] pre-trained word embeddings and dependence-based word embeddings are introduced to represent word features from multiple levels, and the parallel multi-pooling convolutional neural network is used to learn feature representations from different windows in different spaces, thus improving the classification results of event arguments.

(10) He et al. [22] by using dependence-based word embeddings and sentence embeddings as feature inputs, different argument detection frameworks are constructed for simple events and complex events respectively. Each candidate argument is classified as a specific event role by multi-level attention, which improves the performance of event extraction.

(11) Zhao et al. [18] proposes a reinforcement learning-based approach for extracting multiple biomedical events, incorporating pre-trained word embeddings and knowledge base embeddings to optimize the extraction process.

(12) Shen et al. [19] proposes a joint event extraction method based on CNN-BiGRU and attention mechanism, aimed at extracting deeper and more comprehensive features more effectively to accomplish the task.

(13) Gong et al. [20] proposes a document-level joint biomedical event extraction model named BGHGCN, which consists of Bi-directional Long Short-Term Memory (BiLSTM), improved BiAffine Graph Parser (IBGP), GCN and hypergraph convolutional networks (HGCN).

3.2. Results and analysis

3.2.1. General comparison

The event extraction results of different models on the MLEE dataset are shown in Table 4. It can be observed from Table 4 that:

(1) Pyysalo et al. [6] and Zhou et al. [7] using artificial designed features combined with SVM to extract events from biomedical texts has significant performance in traditional rule-based methods and machine learning methods. However, neural network-based methods generally have better extraction results. For example, the F1 score of the model in this paper is better than Pyysalo et al. [6] and Zhou et al. [7] increased by 7.30% and 5.09%, Zhao et al. [14] the F1 score is higher than that of Pyysalo [6] and Zhou et al. [7] this result also proves that the model based on neural network is more suitable for the task of event extraction because it can better mine the latent semantic information in the text, and avoids the artificial design of responsible feature representation.

(2) Li et al. [13], He et al. [10], Zhao et al. [14], Li et al. [12], Gong et al. [20], and the model proposed in this paper all introduce dependency relationships. Among them, He et al. [10] and Li et al. [12] only use dependency-based word embeddings, while Li et al. [13], Zhao et al. [14], and Gong et al. [20] use dependency relationships to extract potential information within relations. The model proposed in this paper integrates dependency information using syntactic distance based on dependency relationships. Li et al. [12]'s

Table 4
Event extraction results of different models.

Model	P(%)	R(%)	F1(%)
Pyysalo et al.	62.28	49.56	55.20
Zhou et al.	55.76	59.16	57.41
Björne et al.	60.65	48.58	53.95
Li et al.	60.77	54.03	57.20
Yan et al.	57.23	54.03	55.58
Zhu et al.	66.08	55.02	60.05
He et al.	90.24	44.50	59.61
Zhao et al.	63.91	56.08	59.74
Li et al.	67.23	53.61	59.65
He et al.	91.05	44.68	59.94
Zhao et al.	66.72	57.83	61.96
Shen et al.	67.92	54.83	60.68
Gong et al.	67.34	58.32	62.46
text	71.90	55.20	62.50

F1 score is 5.70% higher than that of Björne et al. [8], indicating that dependency relationships can lead to gains in event extraction. Zhao et al. [14]’s F1 score is 0.13% higher than that of He et al. [10] and 0.09% higher than that of Li et al. [12]. The reason for this difference may be that mining potential features based on dependencies are more effective than using dependency-based word embeddings. However, among all these models, the model proposed in this paper achieves the optimal performance. Our model’s F1 score is 2.76% higher than that of Zhao et al. [14] and 0.04% higher than that of Gong et al. [20]. This indicates that our model focuses on more important information using syntactic distance, avoiding interference from redundant information.

(3) He et al. [22], Zhao et al. [14], He et al. [10], Shen et al. [19] and the model in this paper is modeled by multi-level attention. Compared with Zhao et al. [14], He et al. [10], He et al. [22], and Shen et al. [19], the F1 scores of the model in this paper are improved by 2.76%, 2.56%, 2.89%, and 1.82%, respectively. He et al. [22] focus on semantic features at the word level and at the candidate pair level, and utilize a sentence-level attention mechanism to mine potential correlations between candidate pairs. Zhao et al. [14] enrich the contextual representation by incorporating document-level sentence information into a hypergraph. Shen et al. [19] utilize an attention mechanism to integrate the feature extraction results of Convolutional Neural Networks (CNN) and Bidirectional Gated Recurrent Units (BiGRU) neural networks. The model not only pays attention to the important context information at the word level and the latent semantic features between candidate pairs, but also introduces the related argument role prototype list as an auxiliary decision-making, thus mining the role-aware feature representation between candidate pairs and argument role. Compared with the above methods, the model has a richer feature representation.

(4) Zhao et al. [18] employ biomedical knowledge bases and reinforcement learning for extracting multiple biomedical events at the sentence level. Their F1 score is 8.01% higher than that of Björne et al. [8], indicating that their model considers additional contextual correlations between multiple biomedical events, and introducing external knowledge bases for word representation can enhance the performance of biomedical event extraction. The F1 score of our model is 0.04% higher than that of Zhao et al. [18], indicating that adopting a question-answering approach can leverage additional question information and recent advances in question-answering to improve model performance.

(5) The model proposed in this paper achieves 62.50% F1 score, which is higher than the F1 score of other advanced models in Table 4. Moreover, the three indicators of the model in this paper are not much different, indicating that the model in this paper has strong generalization and fitting ability. This is mainly due to the following factors: 1) The candidate argument layer proposed in this paper effectively filters a large number of negative samples, thus avoiding the fuzziness of the model to the prediction samples. 2) This paper alleviates the impact of insufficient data by splitting multiple events in a sentence into multiple questions. 3) Efficient post-processing methods filter a large number of invalid event combinations. Although the accuracy of the model in this paper is significantly lower than He et al. [22] model, which may be due to the fact that He et al. [22] train simple and complex events separately, which effectively improves accuracy. But this method needs to distinguish complex events from simple events manually, which is lack of universality.

3.2.2. A detailed comparison of different events

In order to analyze the event extraction performance of the model proposed in this paper on all events, Table 5 reveals the detailed experimental results of Li et al. [12] and the model in this paper on all 19 event types. Since the detailed performance reports are not provided in the original paper for the other models, they are not compared. As can be seen from Table 5, (1) in terms of accuracy P, the model in this paper performs well on 12 event types, while Li et al. [12] only ranks first on 8 event types. Especially in the event types of “Death” and “Breakdown”, the P-scores of this paper’s model are 21.05% and 5.83% higher than those of Li et al. [12]. (2) In terms of recall R, the model in this paper outperforms the model proposed by Li et al. [12] in 11 event types. In the event types of “Synthesis”, “Remodeling” and “Catabolism”, the model in this paper detected 50.00%, 10.00% and 13.33% more valid events than Li et al. [12]. (3) In terms of the F1 scores, the model in this paper has the optimal performance in 14 out of the 19 event types. The F1 scores of this paper’s model in “Binding”, “Positive Regulation” and “Catabolism” event types are higher than those of Li et al. [12]. Li et al. [12] by 8.14% 12.42% and 16.67% The event extraction results on 19 event types demonstrate the effectiveness of the proposed model in handling simple events.

Table 5
Detailed event extraction results of different models on 19 event types.

Event type	Li et al. [12]			Text		
	P	R	F1	P	R	F1
Cell proliferation	100.00	68.89	81.58	97.06	73.33	83.54
Development	86.81	74.53	80.20	87.36	77.57	82.18
Blood Vessel Development	99.27	90.16	94.50	86.32	85.48	85.90
Death	65.62	55.26	60.00	86.67	68.42	76.47
Breakdown	87.50	60.87	71.79	93.33	58.33	71.80
Remodeling	83.33	50.00	62.50	100.00	60.00	75.00
Growth	95.65	73.33	83.02	92.45	81.67	86.73
Synthesis	100.00	50.00	66.67	100.00	100.00	100.00
Gene Expression	90.40	71.07	79.58	90.15	73.91	81.23
Transcription	100.00	23.08	37.50	50.00	12.50	20.00
Catabolism	100.00	20.00	33.33	100.00	33.33	50.00
Phosphorylation	100.00	100.00	100.00	50.0	25.00	33.33
Dephosphorylation	0.00	0.00	0.00	0.00	0.00	0.00
Localization	70.99	68.38	69.66	81.13	63.24	71.07
Binding	61.29	39.68	48.17	72.50	46.03	56.31
Regulation	42.96	27.16	33.28	49.19	25.42	33.51
Positive Regulation	45.92	39.65	42.56	63.57	48.32	54.98
Negative Regulation	51.01	38.49	43.87	60.00	47.48	53.01
Planned Process	44.03	37.36	40.42	50.00	33.85	40.37

As can be seen from Table 5: (1) In terms of accuracy P, the model in this paper performs well on 12 event types, while Li et al. [12] only rank first on 8 event types. In particular, the P-scores of the model in this paper are 21.05% and 5.83% higher than those of Li et al. [12] in the event types “Death” and “Breakdown”. (2) In terms of recall rate R, the proposed model is better than Li et al. [12] in 11 types of events. In the event types of “Synthesis”, “Remodeling” and “Catabolism”, the model in this paper detected 50.00%, 10.00% and 13.33% more valid events, respectively, than Li et al. [12]; (3) In terms of F1 score, the model has the best performance in 14 of 19 event types, and The F1 scores of this model for the event types “Binding”, “Positive Regulation” and “Catabolism” are 8.14%, 12.42% and 16.67% higher than those of Li et al. [12]. Event extraction results on 19 event types demonstrate the effectiveness of the model proposed in this paper in handling simple events.

The performance of the proposed model on nested events is further analyzed. Nested events are considered to be the most challenging case in event extraction and are very common in biomedical applications. In MLEE, there are four types of events (i.e., “Regulation”, “Positive Regulation”, “Negative Regulation” and “Planned Process”) whose parameters may be another event, resulting in nested structure. On the nested events of “Positive Regulation” and “Negative Regulation”, the three evaluation measures of the proposed model are all optimal. The mean value of three of the evaluation indicators was 10.98% higher than Li et al. [12]. In addition, in the “Regulation” event, although the R score of the model is higher than that of Li et al. [12], P and F1 scores were lower than those of Li et al. [12]. These performance comparison results also prove that the proposed model has advantages in dealing with complex events.

3.2.3. Impact of different modules on the model

In order to validate the effect of the individual modules proposed in this paper on the overall model, this study designed sub-models consisting of different modules and evaluated their effectiveness. Table 6 demonstrates the composition and effectiveness of each sub-model. Among them, the reading comprehension paradigm sub-model directly considers argument detection as a relational extraction task for machine reading comprehension; the reading comprehension approach plus candidate argument layer sub-model adds a candidate argument generation layer on top of the reading comprehension paradigm; the reading comprehension approach plus candidate argument layer plus word-level attention sub-model adds word-level contextual attention; the reading comprehension approach plus candidate argument layer plus word-level attention plus candidate pair attention submodel adds the candidate pair attention layer; the reading comprehension approach plus candidate argument layer plus multi-feature attention submodel adds all the attention mechanisms proposed in this paper; the reading comprehension approach plus candidate argument layer plus multi-feature attention plus post-processing is the complete model proposed in this paper.

(1) In order to verify the effectiveness of the candidate argument generation layer, it can be seen from Table 6 that the model using the reading comprehension paradigm and candidate argument layer performs better than the model using only the reading comprehension paradigm, and the F1 score is improved by 6.90%, which also verifies that the candidate argument layer proposed in this paper can effectively filter out irrelevant negative sample information and improve the generalization ability of the model.

(2) In order to verify the advantages of the multi-feature attention mechanism, it can be seen from Table 6 that the F1 score of the reading comprehension paradigm plus candidate argument layer plus word-level attention is 0.40% higher than that of the model using only the reading comprehension paradigm plus candidate argument layer. This proves the advantage of the word-level attention mechanism, that is, the syntactic distance in the dependency tree provides key information for the changing context, so that the keyword information can better help the model pay attention to the relationship between the trigger word and the candidate argument under the same event. On the basis of reading comprehension paradigm + candidate argument level + word-level attention, the F1 score increased by 0.50% after adding candidate pair attention. This shows that by amplifying the similarity

Table 6
Effects of different modules on the model.

Model	P(%)	R(%)	F1(%)
Reading comprehension paradigm + candidate argument level + multi-feature attention + post-processing	71.90	55.20	62.50
Reading comprehension paradigm + candidate argument level + multi-feature attention	69.70	54.00	60.80
Reading comprehension paradigm + candidate argument level + word-level attention + candidate pair attention	71.50	52.30	60.40
Reading comprehension paradigm + candidate argument level + word-level attention	71.50	51.50	59.90
Reading Comprehension Paradigm + Candidate Argument Level	68.60	52.50	59.50
reading comprehension paradigm	70.00	42.10	52.60

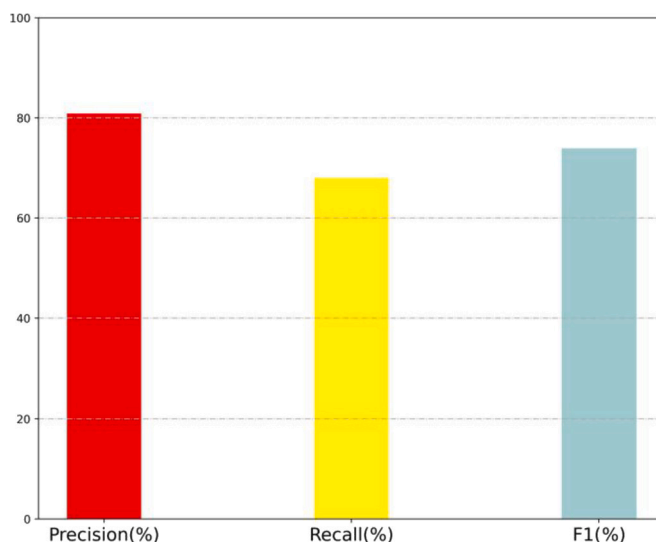


Fig. 5. Experimental results of annotated triggers as test dataset.

and association between candidate pairs can help the model identify the correct event structure. Finally, we find that the F1 score of reading comprehension paradigm + candidate argument layer + multi-feature attention is 0.40% higher than that of reading comprehension paradigm + candidate argument layer + word-level attention + candidate pair attention, which proves that the association information between argument roles and candidate pairs is important for argument extraction.

(3) In order to prove the effectiveness of the post-processing paradigm proposed in this paper, it can be seen from Table 6 that the reading comprehension paradigm + candidate argument layer + multi-feature attention + post-processing has better performance than the reading comprehension paradigm + candidate argument layer + multi-feature attention, and the F1 value is 1.70% higher. It can be seen that although the reading comprehension paradigm + candidate argument layer + multi-feature attention has the ability to obtain all the argument information in an event, there may be some invalid relations among them, which may lead to the event structure directly composed not meeting the expectations. The post-processing paradigm proposed in this paper can effectively avoid these errors and find the most matching event structure under the existing structure.

3.2.4. Error and limitation analysis

Based on the event extraction results of the proposed model, the main reasons for extraction errors are analyzed as follows:

(1) Lack of training data: Some events have few instances in the training set compared to their occurrences in the test set. For example, the “Dephosphorylation” event has only 3 instances in the entire MLEE dataset. As shown in Table 5, the “Dephosphorylation” event was not correctly extracted.

(2) Error propagation in trigger word recognition: Due to the pipeline extraction approach adopted, errors in trigger word recognition severely impact the final event extraction results. Experimental results using annotated trigger words as inputs for the test set, as depicted in Fig. 5, show precision, recall, and F1 scores of 80.90%, 68.00%, and 73.90%, respectively. Compared to using the predictions from Chapter 3 as inputs, these scores are higher by 9.00%, 12.80%, and 11.40%, further confirming the impact of errors in trigger word recognition.

(3) Data Ambiguity: Some words are marked as trigger words in certain contexts, while in others, they are not. Additionally, some entities or trigger words are sometimes labeled as event arguments, while in other cases, they are not. However, these entities or trigger words share the same parsing structure in the sentence. For example, as shown in Fig. 6, sentences S1, S2, S3, and S4 illustrate this situation. The word “expression” serves as a trigger word for the “GeneExpression” event type in S1, but does not trigger any event in S2. In S3, the gene or gene product VEGF is in a Theme role relationship with the event-triggering word “expression”, but in S4, there is no relationship between VEGF and “expression”.

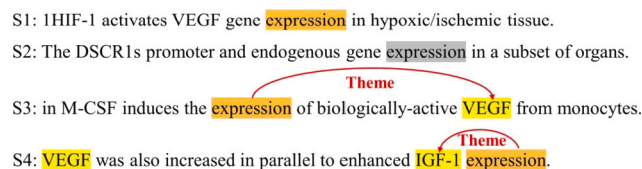


Fig. 6. Data ambiguity instan.

4. Conclusions and future work

In this paper, we propose a method for biomedical event argument detection based on multi-feature fusion and question answering paradigm. Firstly, it divides each event into independent question answering forms, and customizes the corresponding question template for each event, so as to reduce the detection difficulty and event ambiguity while utilizing additional information. Secondly, the argument prior template of each event type is found by using the syntactic distance and the external prior knowledge from the training documents, so as to retain the qualified candidate arguments and reduce the interference of irrelevant arguments. Then, the multi-feature attention mechanism is introduced to fully mine the potential semantic representations between trigger words and candidate arguments, trigger word-candidate argument pairs, and trigger word-candidate argument pairs and roles to complete local and global fine-grained interaction and prevent key semantic loss. Finally, the corresponding argument and its role information of each event are answered by combining multiple features. After the event argument detection, a complete biomedical event is formed according to the proposed post-processing rules. Experimental results on MLEE corpus show that candidate argument generation method, multi-feature attention, machine reading comprehension paradigm and the proposed post-processing rules all effectively improve the model performance, and the proposed model achieves 62.50% F1-score, which is better than other baseline models. This paper focuses on sentence-level biomedical event extraction. However, existing biomedical event corpora suffer from class imbalance issues, such as only 3 events of the type “Dephosphorylation” in the entire corpus. To address this problem, future research could consider introducing external corpora or employing transfer learning methods to alleviate class imbalance. In the real world, more event information is distributed in multiple sentences. Therefore, in the future, the sentence-level event extraction can be extended to the document-level to better capture the cross-sentence event information.

CRedit authorship contribution statement

Jinghan Tian: Writing – review & editing, Writing – original draft, Visualization, Methodology, Formal analysis, Data curation, Conceptualization. **Shuai Xing:** Writing – review & editing, Validation, Supervision, Software. **Qianmin Su:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

Declaration of competing interest

The authors declare that there are no conflict of interests, we do not have any possible conflicts of interest.

Data availability statement

The MLEE dataset used in this study was obtained from the Biomedical Literature and Entity Extraction (MLEE) database, which is publicly available at [<https://nactem.ac.uk/MLEE/>]. The dataset contains annotated biomedical text data, including event triggers, arguments, and their relationships. Researchers interested in accessing the dataset can download it from the provided link. The dataset is distributed under the Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. Please refer to the license terms for details on data usage rights and restrictions.

Acknowledgements

This fund is supported by the Technical Standards Project of Shanghai Science and Technology Innovation Action Plan (21DZ2203100) and the National Natural Science Foundation of China (62006150).

References

- [1] Xinglong Wang, Iain McKendrick, Ian Barrett, Ian Dix, Tim French, Jun'ichi Tsujii, Sophia Ananiadou, Automatic extraction of angiogenesis bioprocess from text, *Bioinformatics* 27 (19) (2011) 2730–2737.
- [2] Tomoko Ohta, Sampo Pyysalo, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Sophia Ananiadou, Jun'ichi Tsujii, Overview of the pathway curation (pc) task of bionlp shared task 2013, in: *Proceedings of the BioNLP Shared Task 2013 Workshop*, 2013, pp. 67–75.
- [3] Hui Liu, Jun Xu, Yan Wu, Qiang Guo, Bulat Ibragimov, Lei Xing, Learning deconvolutional deep neural network for high resolution medical image reconstruction, *Inf. Sci.* 468 (2018) 142–154.
- [4] Emily K. Mallory, Ce Zhang, Christopher Ré, Russ B. Altman, Large-scale extraction of gene interactions from full-text literature using deeptide, *Bioinformatics* 32 (1) (2016) 106–113.

- [5] Maya Rotmensch, Yoni Halpern, Abdulhakim Tlimat, Steven Horng, David Sontag, Learning a health knowledge graph from electronic medical records, *Sci. Rep.* 7 (1) (2017) 5994.
- [6] Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun'ichi Tsujii, Sophia Ananiadou, Event extraction across multiple levels of biological organization, *Bioinformatics* 28 (18) (2012), i575–i581.
- [7] Deyu Zhou, Dayou Zhong, A semi-supervised learning framework for biomedical event extraction based on hidden topics, *Artif. Intell. Med.* 64 (1) (2015) 51–58.
- [8] Jari Björne, Tapio Salakoski, Biomedical event extraction using convolutional neural networks and dependency parsing, in: *Proceedings of the BioNLP 2018 Workshop*, 2018, pp. 98–108.
- [9] Shankai Yan, Ka-Chun Wong, Context awareness and embedding for biomedical event extraction, *Bioinformatics* 36 (2) (2020) 637–643.
- [10] Xinyu He, Lishuang Li, Xingchen Song, Degen Huang, Fuji Ren, Multi-level attention based blstm neural network for biomedical event extraction, *IEICE Trans. Inf. Syst.* 102 (9) (2019) 1842–1850.
- [11] Lixing Zhu, Haoran Zheng, Biomedical event extraction with a novel combination strategy based on hybrid deep neural networks, *BMC Bioinform.* 21 (2020) 1–12.
- [12] Lishuang Li, Yang Liu, Meiyue Qin, Extracting biomedical events with parallel multi-pooling convolutional neural networks, *IEEE/ACM Trans. Comput. Biol. Bioinform.* 17 (2) (2018) 599–607.
- [13] Diya Li, Lifu Huang, Heng Ji, Jiawei Han, Biomedical event extraction based on knowledge-driven tree-lstm, in: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 1421–1430.
- [14] Weizhong Zhao, Jinyong Zhang, Jincai Yang, Tingting He, Huifang Ma, Zhixin Li, A novel joint biomedical event extraction framework via two-level modeling of documents, *Inf. Sci.* 550 (2021) 27–40.
- [15] Kung-Hsiang Huang, Mu Yang, Nanyun Peng, Biomedical event extraction with hierarchical knowledge graphs, *arXiv preprint*, arXiv:2009.09335, 2020.
- [16] Lishuang Li, Jia Wan, Jieqiong Zheng, Jian Wang, Biomedical event extraction based on gru integrating attention mechanism, *BMC Bioinform.* 19 (9) (2018) 93–100.
- [17] Xing David Wang, Leon Weber, Ulf Leser, Biomedical event extraction as multi-turn question answering, in: *Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis*, 2020, pp. 88–96.
- [18] Weizhong Zhao, Yao Zhao, Xingpeng Jiang, Tingting He, Fan Liu, Ning Li, Efficient multiple biomedical events extraction via reinforcement learning, *Bioinformatics* 37 (13) (2021) 1891–1899.
- [19] Chao Shen, Jianhua Tao, Peng Li, Zhao Lv, Guohua Yang, Joint event extraction based on CNN-BiGRU and attention mechanism, in: *2022 Asia Conference on Algorithms, Computing and Machine Learning (CACML)*, IEEE, 2022, pp. 492–497.
- [20] Jinghao Gong, Qinghan Lai, Song Liu, Document-level joint biomedical event extraction model using hypergraph convolutional networks, in: *2022 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, IEEE, 2022.
- [21] Dan Klein, Christopher D. Manning, Accurate unlexicalized parsing, in: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, 2003, pp. 423–430.
- [22] Xinyu He, Ping Tai, Hongbin Lu, Xin Huang, Yonggong Ren, A biomedical event extraction method based on fine-grained and attention mechanism, *BMC Bioinform.* 23 (1) (2022) 308.