

Original Paper

Conditional Probability Joint Extraction of Nested Biomedical Events: Design of a Unified Extraction Framework Based on Neural Networks

Yan Wang¹, PhD; Jian Wang¹, PhD; Huiyi Lu², PhD; Bing Xu², PhD; Yijia Zhang³, PhD; Santosh Kumar Banbhrani¹, PhD; Hongfei Lin¹, PhD

¹School of Computer Science and Technology, Dalian University of Technology, Dalian, China

²Department of Pharmacy, The Second Affiliated Hospital of Dalian Medical University, Dalian, China

³School of Information Science and Technology, Dalian Maritime University, Dalian, China

Corresponding Author:

Jian Wang, PhD

School of Computer Science and Technology

Dalian University of Technology

No 2 Linggong Road

Dalian, 116024

China

Phone: 86 13604119266

Email: wangjian@dlut.edu.cn

Abstract

Background: Event extraction is essential for natural language processing. In the biomedical field, the nested event phenomenon (event A as a participating role of event B) makes extracting this event more difficult than extracting a single event. Therefore, the performance of nested biomedical events is always underwhelming. In addition, previous works relied on a pipeline to build an event extraction model, which ignored the dependence between trigger recognition and event argument detection tasks and produced significant cascading errors.

Objective: This study aims to design a unified framework to jointly train biomedical event triggers and arguments and improve the performance of extracting nested biomedical events.

Methods: We proposed an end-to-end joint extraction model that considers the probability distribution of triggers to alleviate cascading errors. Moreover, we integrated the syntactic structure into an attention-based gate graph convolutional network to capture potential interrelations between triggers and related entities, which improved the performance of extracting nested biomedical events.

Results: The experimental results demonstrated that our proposed method achieved the best F1 score on the multilevel event extraction biomedical event extraction corpus and achieved a favorable performance on the biomedical natural language processing shared task 2011 Genia event corpus.

Conclusions: Our conditional probability joint extraction model is good at extracting nested biomedical events because of the joint extraction mechanism and the syntax graph structure. Moreover, as our model did not rely on external knowledge and specific feature engineering, it had a particular generalization performance.

(*JMIR Med Inform* 2022;10(6):e37804) doi: [10.2196/37804](https://doi.org/10.2196/37804)

KEYWORDS

nested biomedical event; joint extraction; graph convolutional network; GCN; Dice loss; syntactic structure

Introduction

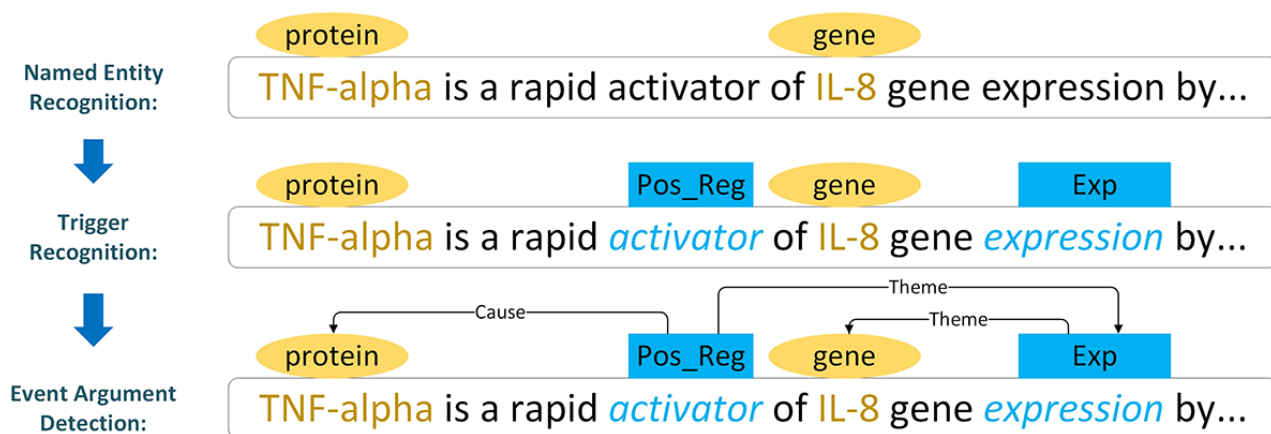
Background

In recent years, event extraction research has attracted wide attention, especially in biomedical event extraction, which is critical for understanding the biomolecular interactions described in the scientific corpus. Events are important concepts in the field of information extraction. However, researchers have different definitions of events, based on different research purposes and perspectives. In the general domain, an event is a specific thing that describes a state change involving different participants, such as the evaluation of automatic content extraction, in which 8 categories and 33 subcategories of events

are defined in a hierarchical structure, and each type of event contains a different semantic role. In the biomedical field, McDonald et al [1] defined event extraction as multirelationship extraction, the purpose of which was to extract semantic role information between different entities in an event. For example, the biomedical natural language processing (BioNLP) evaluation task defined 9 different categories of biochemical events. Each event included an event trigger and at least one event argument, and the different event types had different semantic roles. Unlike the events in automatic content extraction, biomedical events may have nested event phenomena.

To clearly describe the progress of biomedical event extraction, we defined 4 concepts for biomedical events, as shown in Figure 1 and Textbox 1.

Figure 1. Basic progress of biomedical event extraction, where yellow boxes represent the type of entity and the blue boxes represent the type of trigger. Theme and cause represent the relationship between participant and event, namely, argument detection. IL-8: interleukin 8; TNF-alpha: tumor necrosis factor.



Textbox 1. Concepts for biomedical events.

Event type
The semantic type of different events
Event description
A complete sentence or clause in the text that specifically describes at least one event
Event trigger
A word or phrase representing the occurrence of an event in the event description; usually of a <i>verb</i> or <i>nonverb</i> nature, and its category is event type; it should be noted that each event has only 1 event trigger.
Event argument
The event participants describe the different semantic roles in the event, whose type represents the relationship between the event and related participants; in the biomedical event system, there are 6 different semantic roles, where “theme” and “cause” are core arguments.

The task of event extraction comprises 3 subtasks: named entity recognition, trigger recognition, and event argument detection. Previous studies have relied on pipeline methods [2-5] to extract biomedical events. For example, given the event description (a sentence) shown in Figure 1, the event extraction system can find 2 entities (“TNF-alpha” and “IL-8”) in this sentence at the named entity recognition step. After recognizing triggers, it can identify a *positive regulation* (“Pos_Reg”) event mention triggered by a word *activator* and an *expression* (“Exp”) event mention triggered by a word *expression*. On the basis of the recognized entities and triggers, the system detects arguments

and associates them with the related event triggers. Thus, the entity “TNF-alpha” is a participant in the *positive regulation* event, and the entity “IL-8” is a participant in the *expression* event. As the result of the previous step is the input of the subsequent step, the pipeline methods probably introduce cascading errors if the precision of the previous step is biased.

As the syntactic dependency tree enriches the feature representation, previous studies tended to use syntactic relations to improve the performance of event extraction. For example, Kilicoglu et al [2] leveraged external tools to segment sentences, annotate parts of speech (POS), and parse syntactic dependency.

Then, they joined these features to extract biomedical events using a dictionary and rules. Björne et al [4] transferred the syntactic relations to the path embeddings, then combined them with word embeddings, POS embeddings, entity embeddings, distance embeddings, and relative position embeddings to feed into the convolutional neural network (CNN) model to extract biomedical events. However, the previous studies only adopted syntactic relations as the external features and ignored the interrelations between triggers and related entities obtained from the syntactic dependency tree, which improved the performance of extracting simple events but not nested events.

In this study, we mainly used the multilevel event extraction (MLEE) corpus [6] and the BioNLP shared task (BioNLP-ST) 2011 Genia event (GE) corpus [7] to evaluate our method. There is some explanation regarding the MLEE extending event extraction methods to the biomedical information field and covering all levels of biological tissue from molecules to entire organisms. The MLEE label scheme is the same as the BioNLP event system but has more abundant event types: 4 major categories (anatomical, molecular, general, and planned) and 19 subcategories. The specific information is shown in Table 1.

Table 1. Primary event types and argument roles in the multilevel event extraction corpus (N=6827).

Event and subevent types	Core arguments	Values, n (%)
Anatomical		
Cell proliferation	Theme (entity)	133 (2.42)
Development	Theme (entity)	316 (4.81)
Blood vessel development	Theme (entity)	855 (12.91)
Growth	Theme (entity)	469 (2.65)
Death	Theme (entity)	97 (1.53)
Breakdown	Theme (entity)	69 (1.1)
Remodeling	Theme (entity)	33 (0.45)
Molecular		
Synthesis	Theme (entity)	17 (0.3)
Gene expression	Theme (entity)	435 (6.66)
Transcription	Theme (entity)	37 (0.61)
Catabolism	Theme (entity)	26 (0.39)
Phosphorylation	Theme (entity)	33 (0.5)
Dephosphorylation	Theme (entity)	6 (0.09)
General		
Localization	Theme (entity)	450 (6.87)
Binding	Theme (entity)	187 (2.92)
Regulation	Theme (entity or event) and cause (entity or event)	773 (11.81)
Positive regulation	Theme (entity or event) and cause (entity or event)	1327 (20.33)
Negative regulation	Theme (entity or event) and cause (entity or event)	921 (14.08)
Planned		
Planned process	Theme (entity or event)	643 (9.9)

To abate the impact of cascading errors, we propose an end-to-end conditional probability joint extraction (CPJE) method that can effectively transmit trigger distribution information to the event argument detection task. To capture the interrelations between triggers and related entities and improve the performance of extracting nested biomedical events, we integrated the syntactic dependency tree into an attention-based gate graph convolutional network (GCN), which can capture the flow direction of the key information. The contributions of this study are as follows:

1. We propose an end-to-end CPJE framework, CPJE, which effectively leverages trigger distribution information to

enhance the performance of event argument detection and weakens cascading errors in the overall event extraction process.

2. We used the syntactic dependency tree to capture the interrelations between triggers and related entities and integrated the tree into an attention-based gate GCN to extract nested biomedical events.
3. We obtained state-of-the-art performance on the MLEE and BioNLP-ST 2011 GE corpora for extracting nested biomedical events.

We summarize the current frameworks for event extraction tasks in the *Related Works* section. We introduce our framework

in the *Methods* section. We display the overall performance in the *Results* section. We present the ablation study, visualization, and case study in the *Discussion* section. We summarize this work and discuss future research directions in the *Conclusions* section.

Related Works

The biomedical event extraction problem is similar to general domain event extraction and entity relationship extraction; therefore, we have many theoretical foundations and experimental methods that can be used for reference.

Entity Relationship Extraction

Biomedical events can be regarded as complex relationship extraction tasks, and relationship extraction methods have achieved excellent results in various fields. Therefore, we studied some relationship extraction methods to help conceive the construction of event extraction models. With the development of deep learning, an increasing number of researchers have used deep learning algorithms to achieve the joint extraction of entity relationships [8]. To solve the problem of a sparse number of labeled samples, distant supervision methods have been applied to the relationship extraction task [9]. Deep reinforcement learning (RL) algorithms have also been applied to the relationship extraction task to solve noisy data samples [10]. In addition, with the widespread application of graph neural networks (GNNs), GCNs have been used in certain relation-extraction tasks [11,12].

General Domain Event Extraction

In general, news event extraction is a research hot spot. Some methods have improved the performance of event extraction by studying feature engineering. Sentence-level feature extraction included combinational features of triggers and event arguments [13] or combinational features of triggers and entity relationships [14]. Document-level feature extraction included common information event extraction from multiple documents [15] and joint event argument extraction based on latent-variable semi-Markov conditional random fields [16]. Others have also used deep learning to reduce feature engineering, which improves a model's generalization ability and extraction performance; for example, learning context-dependency information with recurrent neural networks [17], detecting events with nonconsecutive CNNs [18], and obtaining syntactic structure information with GCNs [19]. All these methods have laid a better foundation for the extraction of biomedical events.

Biomedical Event Extraction

Extracting biomedical events is one of the BioNLP-STs [7,20,21]. Previous studies mainly explored human-engineered features based on a support vector machine model [22-25]. Owing to error transmission in the pipeline approach, Riedel et al [26] developed a joint model with dual decomposition, and Venugopal et al [27] leveraged Markov logic networks for joint inference. Recently, most studies have observed remarkable benefits of neural models. For example, some have started to add POS tags and syntactic parsing with different neural models [28], improved the biomedical event extraction model using

semisupervised frameworks [29], attempted to use attention mechanisms to obtain the semantic relationship of biomedical texts [5], and used distributed representations to obtain context embedding [3,4,30,31]. To incorporate more information from the biomedical knowledge base (KB), Zhao et al [32] leveraged a RL framework to extract biomedical events with representations from external biomedical KBs. Li et al [33] fused gene ontology into tree long short-term memory (LSTM) models with distributional representations. Huang et al [34] used a GNN to hierarchically emulate 2 knowledge-based views from the Unified Medical Language System with conceptual and semantic inference paths. Trieu et al [35] used multiple overlapping, directed, acyclic graph structures to jointly extract biomedical entities, triggers, roles, and events. Zhao et al [36] combined a dependency-based GCN with a hypergraph to jointly extract biomedical events. Ramponi et al [37] proposed a joint end-to-end framework that regards biomedical event extraction as sequence labeling with a multilabel aware encoding strategy.

Compared with these methods, our approach joint extracts the biomedical events with a probability distribution of triggers, which alleviates the cascading errors introduced by the pipeline methods. Moreover, considering the potential interrelations between triggers and related entities, our approach integrates the syntactic structure into an attention-based gate GCN to capture the flow direction of key information, which greatly improves the extraction performance for nested biomedical events. It is important to mention that our approach does not require any external resources to assist the biomedical event extraction task.

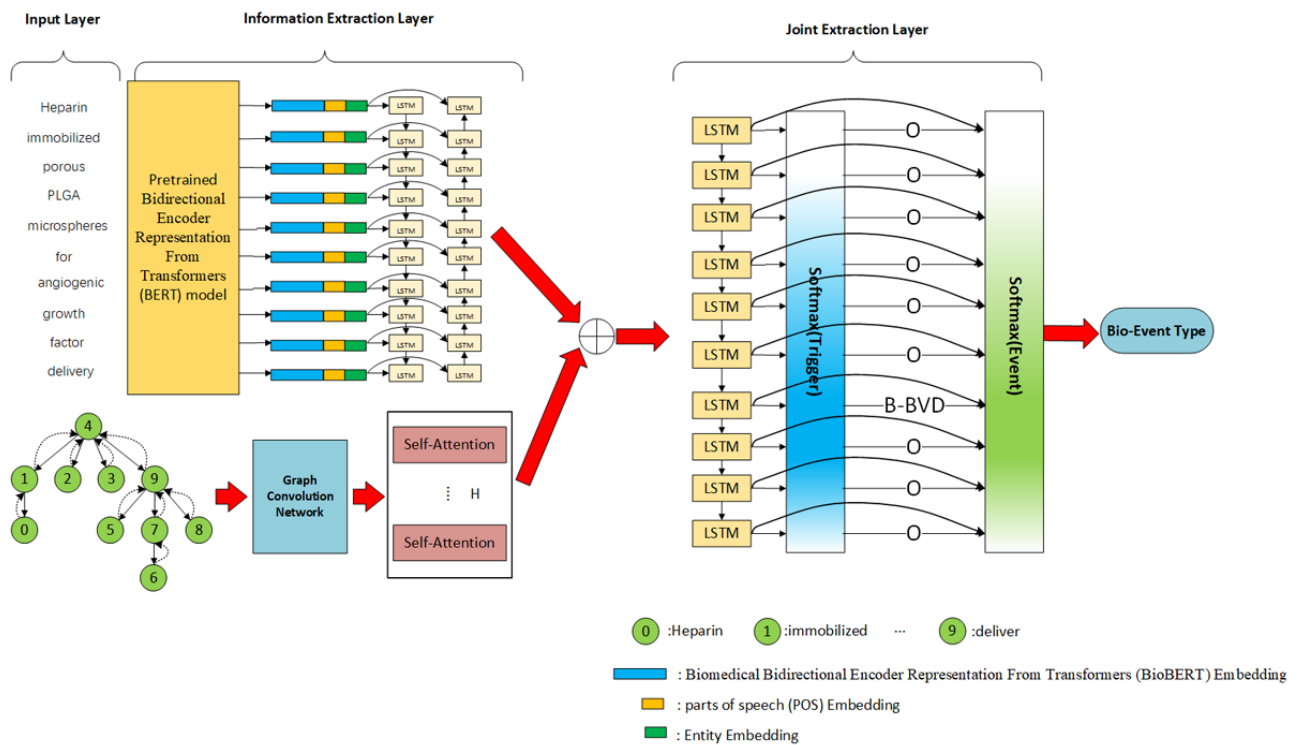
Methods

Overview

This section illustrates the proposed CPJE model. Let $W=\{w_1,w_2,\dots,w_n\}$ be a sentence of length n , where w_i is the i th word in a sentence. Similarly, $E=\{e_1,e_2,\dots,e_k\}$ is a set of entities mentioned in a sentence, where k is the number of entities. As the trigger may comprise multiple tokens, we used the BIO tag scheme to annotate the trigger type of each token in the sentence. When we obtained the corresponding event trigger in the sentence, we used this information to predict the corresponding event arguments.

As shown in Figure 2, our CPJE model mainly includes 3 layers: an input layer, an information extraction layer, and a joint extraction layer. The input layer converts unstructured text information (such as word sequences, syntactic structure trees, POS label representations, and entity label information) into a structured discrete representation and inputs it into the next layer. The information extraction layer converts discrete information into continuous feature representations, which deeply extracts the semantic and dependence information in a sentence. The joint extraction layer parses the previous fusion information and sends the parsed information into the trigger softmax classifier and event softmax classifier to jointly extract biomedical events.

Figure 2. The architecture of the conditional probability joint extraction framework, where numbers 0 to 9 represent each word in the sentence, the blue bar represents BioBERT embedding, the yellow bar represents POS-tagging embedding, and the green bar represents entity embedding. BERT: Bidirectional Encoder Representation From Transformers; BioBERT: Biomedical Bidirectional Encoder Representation From Transformers; B-BVD: B-blood vessel development; LSTM: long short-term memory; POS: parts of speech.



Information Extraction Layer

This is not explained in detail as the input layer was too superficial (only converting the text into a sequence of numbers). Each module of the information extraction layer is presented in the following sections.

Word Representation

In the word representation module, to improve the representation capability of the initial features, each word w_i in the sentence is transformed to a real-valued vector x_i by concatenating the embeddings described in the following sections.

Biomedical Bidirectional Encoder Representation From Transformers Embedding

We used the Biomedical Bidirectional Encoder Representation from Transformers (BioBERT) pretraining model [38] to obtain the dynamic semantic representation of the word w_i . BioBERT embedding comprises token embedding, segment embedding, and position embedding, which is encoded as a consequence by a multilayer bidirectional transformer. Thus, it includes rich semantic and positional information. Furthermore, it can solve the polysemy problem of words. We define a_i as the word vector representation of the word w_i .

POS-Tagging Embedding

We used a randomly initialized POS-tagging embedding table to obtain each POS-tagging vector. We defined b_i as the POS-tagging vector representation of the word w_i .

Entity Label Embedding

Similar to the POS-tagging embedding, we used the BIO label scheme to annotate the entities mentioned in the sentence and convert the entity type label into a real-value vector by consulting the embedding table. We defined c_i as the entity vector representation of the word w_i .

The transformation from the token w_i to the vector x_i converts the input sentence W into a sequence of real-valued vectors $X=\{x_1,x_2,\dots,x_n\}$, $x_i = [a_i \oplus b_i \oplus c_i]$, where $[\oplus]$ is the concatenation operation, x_i is the μ dimension (ie, the sum of the dimensions of a_i , b_i , and c_i), and $X \in \mathbb{R}^{n \times \mu}$. X is fed into the subsequent blocks to obtain more valuable information for extracting biomedical events.

Bidirectional LSTM

To obtain the context information of the input text and avoid the gradient explosion problem caused by long texts, we chose the classic bidirectional LSTM (BiLSTM) structure to extract the context features of the word representations.

We fed the word representation sequence $X=\{x_1,x_2,\dots,x_n\}$ into BiLSTM to obtain the forward hidden unit h_t^f and the backward hidden unit h_t^b with ϕ dimension in time t according to equation 1. We represented all the hidden states of the forward LSTM and backward LSTM as $L_{for} \in \mathbb{R}^{n \times \phi}$ and $L_{back} \in \mathbb{R}^{n \times \phi}$, respectively, where n is the number of LSTM hidden units:

$$\begin{cases} h_t^f = \overrightarrow{LSTM}(h_{t-1}^f \oplus x_t) \\ h_t^b = \overleftarrow{LSTM}(h_{t-1}^b \oplus x_t) \\ L_{for} = \{h_1^f, h_2^f, \dots, h_n^f\} \\ L_{back} = \{h_1^b, h_2^b, \dots, h_n^b\} \end{cases}$$

Finally, we concatenated these 2 matrices to obtain the context representation $L \in \mathbb{R}^{n \times 2\varphi}$ of BiLSTM:

$$L = [L_{for} \oplus L_{back}]$$

Gate GCN

To obtain the syntactic dependence in a sentence, we reference the method proposed by Liu et al [19] to apply a gate GCN model to analyze the sentence-dependent features. We considered an undirected graph $G=(V, \varepsilon)$ as a syntactic dependency tree for the sentence W , where V is the set of nodes and ε is the set of edges. Defining $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ ($|\mathcal{V}| = n$), v_i represents each word w_i of sentence W , and each edge $(v_i, v_j) \in \mathcal{E}$ represents a directed syntactic arc from word w_i to word w_j , with dependency type Re . In addition, for the sake of moving information along the direction, we add the corresponding reversed edge (v_w, v_i) with dependency type Re' and self-loops (v_i, v_i) for any node v_i . According to statistics, we used the Stanford Parser [39] to obtain approximately 50 different kinds of syntactic dependency. To facilitate the GCN internal calculation, we only considered the direction of information flow and simplified the original dependency into 3 forms, as shown in equation 4:

$$Re = \begin{cases} direct, & i! = j \text{ and } (v_i, v_j) \in \mathcal{E} \\ reverse, & i! = j \text{ and } (v_j, v_i) \in \mathcal{E} \\ loop, & i == j \end{cases}$$

For node $v \in \mathcal{V}$, we can use the hidden vector $h_v^{(j)}$ in the j th gate GCN layer to compute the hidden vector $h_v^{(j+1)}$ of the next layer:

$$h_v^{(j+1)} = RELU(\sum_{u \in N(v)} g_{u,v}^{(j)} (W_{Re(u,v)}^{(j)} h_u^{(j)} + b_{Re(u,v)}^{(j)}))$$

where $Re(u,v)$ is the dependency type between nodes u and v , $W_{Re(u,v)}^{(j)}$ and $b_{Re(u,v)}^{(j)}$ are the weight matrix and bias, respectively. $N(v)$ is the set of neighbors of node v , including V . The weight of edge (u, v) is $g_{u,v}^{(j)}$, which applies the gate to the edge to indicate the importance of the edge, as shown in equation 6:

$$g_{u,v}^{(j)} = sigmoid(h_u^{(j)} V_{Re(u,v)}^j + d_{Re(u,v)}^j)$$

Here, $V_{Re(u,v)}^j$ and $d_{Re(u,v)}^j$ are the gate weight matrix and bias, respectively. We used BioBERT embedding $A=\{a_1, a_2, \dots, a_n\}$ to

initialize the input of the first GCN layer. Stacking k GCN layers can obtain a syntactic information matrix $G \in \mathbb{R}^{n \times m}$, where m is the dimension of node v_i with the same dimension of a_i .

Multi-Head Attention

As shown in Figure 2, multi-head attention [40] comprises H self-attentions, which can thoroughly learn the similarity between nodes and calculate the importance of each node so that the model can focus on more critical node features. Let W_i^Q, W_i^K , and W_i^V be the i th initialized weight matrix of Q, K , and V , known by equation 7:

$$\begin{cases} Q_i = G W_i^Q \\ K_i = G W_i^K \\ V_i = G W_i^V \end{cases}$$

Here, $W_i^Q \in \mathbb{R}^{m \times d_k}$, $W_i^K \in \mathbb{R}^{m \times d_k}$, $W_i^V \in \mathbb{R}^{m \times d_v}$, and $d_k=d_v=m/H$.

We calculated the scoring matrix of the i th head according to equation 8. After concatenating H heads, we used equation 9 to obtain the attention output matrix M . $W^O \in \mathbb{R}^{Hd_v \times m}$ is the linear transformation matrix:

$$head_i = softmax(\frac{Q_i K_i^T}{\sqrt{d_k}}) V_i$$

$$M = [head_1 \oplus head_2 \oplus \dots \oplus head_H] W^O$$

Joint Extraction Layer

Tagger

The tagger comprises a unidirectional LSTM that takes the context representation given by BiLSTM as the input and the syntactic dependency representation generated by the attention GCN module to parse the information of the previous layer. Let $T = [M \oplus L], T \in \mathbb{R}^{n \times (m+2\varphi)}$. After the tagger module, we obtained the output matrix O , which was sent to the conditional probability extraction module.

Conditional Probability Extraction

Most joint extraction models input the same source information into different subtask classifiers simultaneously to achieve information sharing, as shown in equation 10, where y_{t_i} is the output of the trigger in time step i and y_{a_j} is the output of the argument in step j .

$$\begin{cases} y_{t_i} = softmax^{tri}(W^{tri} O_i + b^{tri}) \\ y_{a_j} = softmax^{arg}(W^{arg} O_j + b^{arg}) \end{cases}$$

However, when the occurrence frequency of 2 subtasks in the same data set varies significantly, the model easily focuses on high-frequency subtasks and ignores low-frequency subtasks. Similar to the biomedical event extraction task, for the trigger

recognition and event argument detection subtasks, each event trigger (ie, biomedical event) may contain 0, 1, or 2 participating elements, and the participating element may also be another event; therefore, the contribution of the trigger recognition task will be greater than that of the event argument detection task. To alleviate the abovementioned problems and reduce the cascading errors between these 2 subtasks, we combined the softmax output after trigger recognition and the source information to extract the trigger vector Tr_i and event argument vector Can_j according to the location of triggers and candidate arguments. Finally, by aggregating and inputting them into the event extraction classifier and learning the distribution features of the trigger label, our model directly achieved biomedical event extraction without postprocessing.

$$\begin{cases} y_{t_i}, soft_i = softmax^{tri}(W^{tri}O_i + b^{tri}) \\ Tr_i = \frac{1}{m} \sum_{k=i_0}^{i_m} [soft_k \oplus O_k] \\ Can_j = \frac{1}{n} \sum_{k=j_0}^{j_n} [soft_k \oplus O_k] \\ y_{e_{i,j}} = softmax^{event}(W^{event}[Tr_i \oplus Can_j] + b^{event}) \end{cases}$$

Here, W^{tri} and b^{tri} are the weight matrix and bias for trigger recognition, separately. The probability output of the trigger softmax of the k th word is $soft_k$. W^{event} and b^{event} are the weight matrix and bias for event extraction, separately. The number of words of the i th trigger and the j th candidate argument are i_m and j_n , separately. O_k is the source information vector of the k th word.

Comparing equation 10 with equation 11, we found that it only realizes the joint extraction of triggers and event arguments using equation 10; therefore, it needs postprocessing to seek out the tuple of events. However, owing to the aggregation of trigger distribution information, we can discover which event argument belongs to the trigger of step t using equation 11.

Joint Dice Loss

Owing to the sparse data of the biomedical event corpus and the imbalance between positive and negative examples, the cross-entropy or negative log-likelihood loss function causes a large discrepancy between precision and recall. To alleviate this problem, we propose using a joint weight self-adjusting Dice loss function [41], as follows:

$$J(\theta) = \frac{1}{N} \sum_{l=1}^N (\text{mean}(\sum_{i=1}^{n_p} \left(1 - \frac{2(1-p_{t_i})p_{t_i}y_{t_i} + \lambda}{(1-p_{t_i})p_{t_i} + y_{t_i} + \lambda}\right)) + \beta * \text{mean}(\sum_{i=1}^{t_p} \sum_{j=1}^{e_p} (1 - \frac{2(1-p_{e_{i,j}})p_{e_{i,j}}y_{e_{i,j}} + \lambda}{(1-p_{e_{i,j}})p_{e_{i,j}} + y_{e_{i,j}} + \lambda})))$$

Here, N is the number of sentences in the corpus; n_p , t_p , and e_p are the number of tokens, extracted trigger candidates, and arguments of the l th sentence, λ is for smoothing purposes, β is a hyperparameter to adjust the loss, and θ is the model's parameters that should be trained.

Training

The CPJE model was trained using several epochs. In each epoch, we divided the training set into batches, each containing a list of sentences and each sentence containing a set of tokens of variable lengths. One batch was in progress at a time step.

For each batch, we first ran the information extraction layer to generate the context representation $L \in \mathbb{R}^{n \times 2\varphi}$ and the attention representation with syntactic information $M \in \mathbb{R}^{n \times m}$. Then, we combined L and M as the input of LSTM to generate source information O . In the end, we ran the joint extraction layer to compute gradients for overall network output (triggers and events). After that, we back propagated the errors from the output to the input through CPJE and updated all the network parameters. The overall procedure of the CPJE model is summarized in [Textbox 2](#).

Textbox 2. The training procedure of the conditional probability joint extraction model.

Input	
1.	Sequence of tokens $\{w_1, \dots, w_n\}$ along with corresponding event labels
2.	Set of edges $\{e_{12}, \dots, e_{ij}, \dots, e_{mn}\}$ for each corresponding token
Output	
All parameters in the conditional probability joint extraction model	
1.	For each epoch do
2.	For each epoch do
3.	Generate L and M by information extraction layer via equations 3 and 9
4.	Concatenate L and M as T
5.	Generate the source information $O = \{o_1, \dots, o_n\}$ by long short-term memory
6.	Compute the trigger scores y_t and the trigger softmax probability $soft$ by the "SoftMax Trigger" block in the joint extraction layer via the first equation in equation 11
7.	Fuse O and $soft$ via the second and third equations in equation 11
8.	Compute the event scores y_e by the "SoftMax Event" block in the joint extraction layer via the fourth equation in equation 11
9.	Update the parameters by the back propagation algorithm
10.	End for
11.	End for

Data

Our experiments were conducted mainly on the MLEE corpus [6], as shown in Table 2, which has 4 categories containing 19 predefined trigger subcategories. There are 262 documents with 56,588 words in total, with 8291 entities and 6677 events. From Table 2, we note that the number of anatomical-level events is higher than the number of molecular-level and planned-level

events, although general biomedical events dominate overall. Overall, 18% (1202/6677) of the total events involved either direct or indirect arguments at both the molecular and anatomical levels. From Table 1, we find that the arguments of regulation, positive regulation, negative regulation, and planned process events may not be only entities but also other events; therefore, these events are nested events, which account for approximately 54.87% (3664/6677) of all events.

Table 2. The multilevel event extraction statistical information.

Item	Training, n (%)	Development, n (%)	Test, n (%)	Total, N
Document	131 (50)	44 (16.8)	87 (33.2)	262
Sentence	1271 (48.73)	457 (17.52)	880 (33.74)	2608
Word	27,875 (49.26)	9610 (16.98)	19,103 (33.76)	56,588
Entity	4147 (50.02)	1431 (17.26)	2713 (32.72)	8291
Event	3296 (49.36)	1175 (17.6)	2206 (33.04)	6677
Anatomical	810 (48.36)	269 (16.06)	596 (35.58)	1675
Molecular	340 (48.2)	125 (17.7)	240 (34.0)	705
General	1851 (50.66)	627 (17.16)	1176 (32.18)	3654
Planned	295 (45.9)	154 (24.0)	194 (30.2)	643

In addition, we verified our experiment using the BioNLP-ST 2011 GE corpus [7]. As shown in Table 3, the BioNLP-ST 2011 GE corpus defines 9 biomedical event types. It is noted that a *binding* event probably requires >1 protein entity as its theme argument, and a *regulation* event is likely to require a protein

or an event as its theme argument and needs a protein or an event as its cause argument. There were 37.20% (9288/24,967) of events (regulation, positive regulation, and negative regulation) that led to a nested structure.

Table 3. The primary event types and core argument roles in the BioNLP-STa 2011 GEb corpus and the important statistical information of the GE corpus.

Event types and BioNLP-ST 2011 GE items	Core arguments	Values, N
Event type		
Gene expression	Theme (protein)	N/A ^c
Transcription	Theme (protein)	N/A
Protein catabolism	Theme (protein)	N/A
Phosphorylation	Theme (protein)	N/A
Localization	Theme (protein)	N/A
Binding	Theme (protein) ^d	N/A
Regulation	Theme (protein or event) and cause (protein or event)	N/A
Positive regulation	Theme (protein or event) and cause (protein or event)	N/A
Negative regulation	Theme (protein or event) and cause (protein or event)	N/A
BioNLP-ST 2011 GE corpus statistics		
Document	N/A	1224
Word	N/A	348,908
Entity	N/A	21,616
Event	N/A	24,967

^aBioNLP-ST: BioNLP shared task.

^bGE: Genia event.

^cN/A: not applicable.

^dRepresents the number of arguments >1.

Hyperparameter Setting

For the hyperparameter settings of our experiment, we used 768 dimensions for the BioBERT embeddings and set 64 dimensions for the POS-tagging and entity label embeddings. We applied a 1-layer BiLSTM with 128 hidden units and used a 2-layer GCN and 2-head self-attention for our model. The dropout rate was 0.3, the learning rate was 0.01, and the optimization function was stochastic gradient descent (SGD). The training of our CPJE model was based on the operating system of Ubuntu

20.04, using PyTorch (version 1.9.0) and Python (version 3.8.8). The graphics processing unit was an NVIDIA TITAN Xp with 12 GB of memory.

Results

Overall Performance on MLEE

We compare our performance with the baselines shown in [Textbox 3](#).

Textbox 3. Baselines for performance.

<p>EventMine</p> <p>Pyysalo et al [6] applied a pipeline-based event extraction system, mainly relying on support vector machine classifiers to implement trigger recognition and event extraction.</p> <p>Semisupervised learning</p> <p>This is a semisupervised learning framework proposed by Zhou et al [30], which can use unannotated data to extract biomedical events.</p> <p>Convolutional neural network</p> <p>Wang et al [3] used convolutional neural networks and multiple distributed feature vector representations to achieve event extraction tasks.</p> <p>mdBLSTM (bidirectional long short-term memory with a multilevel attention mechanism and dependency-based word embeddings)</p> <p>He et al [5] proposed a bidirectional long short-term memory neural network based on a multilevel attention mechanism and dependency-based word embeddings to extract biomedical events.</p> <p>Reinforcement learning+knowledge bases</p> <p>Zhao et al [32] proposed a framework of reinforcement learning with external biomedical knowledge bases for extracting biomedical events.</p> <p>DeepEventMine</p> <p>Trieu et al [35] proposed an end-to-end neural model. It uses a multioverlapping directed acyclic graph to detect nested biomedical entities, triggers, roles, and events.</p> <p>Hierarchical artificial neural network</p> <p>Zhao et al [36] proposed a 2-level modeling method for document-level joint biomedical event extraction.</p>

Table 4 illustrates the overall performance against the state-of-the-art methods with gold standard entities. As seen in this table, our CPJE model achieved only a slight improvement in the trigger recognition task. For the event extraction task, the F_1 score was significantly better than the other baselines. Notably, the gap between the precision and recall of our model was much smaller than that of the mdBLSTM (bidirectional long short-term memory with a multilevel attention mechanism and dependency-based word embeddings) model, and the precision was much better than that of the RL+KBs model. This

indicates that our model had a better effect on reducing cascading errors than the pipeline models. In addition, the hierarchical artificial neural network (HANN) model was also a joint extraction model; however, its performance is disappointing. This is because the HANN model focuses on extracting document-level biomedical events, which contain many cross-sentence entities, triggers, and events. However, other models aim to extract sentence-level events; therefore, the performance of these models is better than that of the HANN model.

Table 4. Overall performance on multilevel event extraction compared with the state-of-the-art methods with gold standard entities.

Method	Trigger recognition (%)			Event extraction (%)		
	Precision	Recall	F ₁ score	Precision	Recall	F ₁ score
EventMine ^a	70.79	81.69	75.84	62.28	49.56	55.20
SSL ^{a,b}	72.17	82.26	76.89	55.76	59.16	57.41
CNN ^{a,c}	80.92	75.23	77.97	60.56	56.23	58.31
mdBLSTM ^{a,d}	82.79	76.56	79.55	90.24	44.50	59.61
RL ^e +KBs ^{a,f}	N/A ^g	N/A	N/A	63.78	56.81	60.09
DeepEventMine ^h	N/A	N/A	N/A	69.91	55.49	61.87
HANN ^{h,i}	N/A	N/A	N/A	63.91	56.08	59.74
Our model ^h	82.20	78.25	80.18	72.26	55.23	62.80 ^j

^aPipeline model.

^bSSL: semisupervised learning.

^cCNN: convolutional neural network.

^dmdBLSTM: bidirectional long short-term memory with a multilevel attention mechanism and dependency-based word embeddings

^eRL: reinforcement learning.

^fKB: knowledge base

^gN/A: not applicable.

^hJoint model.

ⁱHANN: hierarchical artificial neural network.

^jThe best value compared with baselines.

The Performance for Nested Events on MLEE

To evaluate the effectiveness of our model for improving the nested biomedical event extraction, we split the test set into 2 parts (*simple* and *nested*). *Simple* means that 1 event only regards the entities as its arguments; *nested* means that one of the arguments of an event may be another event. In general, nested events are present in regulation, positive regulation, negative regulation, and planned process events.

Table 5 illustrates the performance (F₁ scores) of the CNN model [3], the RL+KBs model [32], the DeepEventMine [35]

model, the HANN [36] model, and our model in the trigger recognition and event extraction subtasks. In the *simple* and *nested* data of triggers, our framework was 0.44% and 1.25% better than the CNN model, which demonstrates that our model can improve the performance of trigger recognition. However, there is no significant difference between simple and nested triggers. In the *nested* data of events, our model was 6.97% higher than the CNN model, 2.57% higher than the RL+KBs model, 9.53% higher than the DeepEventMine model, and 15.8% higher than the HANN model, which illustrates that our CPJE model of using a gate GCN and an attention mechanism helps to enhance the performance of extracting nested events.

Table 5. The F1 score performance on simple events, nested events, and all events on the multilevel event extraction corpus.

Subtask and model	Simple (%)	Nested (%)	All (%)
Trigger			
CNN ^a	79.52	78.80	78.52
RL ^b +KBs ^c	N/A ^d	N/A	N/A
DeepEventMine	N/A	79.12	N/A
HANN ^e	N/A	N/A	N/A
Our model	79.96 ^f	80.05 ^f	80.18 ^f
Event			
CNN	61.33	54.29	58.87
RL+KBs	N/A	58.69	60.09
DeepEventMine	N/A	51.73	61.87
HANN	77.08 ^f	45.46	59.74
Our model	64.85	61.26 ^f	62.80 ^f

^aCNN: convolutional neural network.

^bRL: reinforcement learning.

^cKB: knowledge base.

^dN/A: not applicable.

^eHANN: hierarchical artificial neural network.

^fThe best value compared with other models.

The Performance for All Events on MLEE

To illustrate the impact of our framework on different events in more detail, [Table 6](#) presents the event extraction performance for all event types. From this table, we obtain the best extraction

performance for dephosphorylation events and the worst performance for transcription events. In addition, the catabolic events had the best extraction precision, and the phosphorylation events had the best extraction recall rate.

Table 6. The extraction performance for different events on multilevel event extraction corpus.

Events	Precision (%)	Recall (%)	F ₁ score (%)
Cell proliferation	62.50	58.57	60.47
Development	51.82	66.43	58.22
Blood vessel development	90.42	72.66	80.57
Growth	78.02	50.58	61.37
Death	79.12	44.32	56.81
Breakdown	71.30	48.30	57.59
Remodeling	85.71	58.32	69.41
Synthesis	48.00	20.30	28.53
Gene expression	74.72	82.42	78.38
Transcription	16.67	33.33	22.22
Catabolism	100.00	50.00	66.67
Phosphorylation	90.00	100.00	94.74
Dephosphorylation	100.00	100.00	100.00
Localization	76.86	49.98	60.57
Binding	74.52	51.23	60.71
Regulation	63.82	51.49	56.99
Positive regulation	78.28	50.66	61.51
Negative regulation	64.35	54.69	59.13
Planned process	69.57	51.86	59.42
All	64.85	61.26	62.80

Overall Performance on BioNLP-ST 2011 GE

To improve persuasion, we extended our experiment to the BioNLP-ST 2011 GE corpus. We compared our event extraction results with those of previous systems using the same corpus, as shown in [Table 7](#). Among them, the Turku Event Extraction System (TEES) [42], EventMine [6], and stacked generalization [25] systems are based on support vector machines with designed features. The TEES-CNNs [4] are CNNs integrated into the TEES system to extract relations and events. The DeepEventMine [35] is based on bidirectional transformers and an overlapping directed acyclic graph to jointly extract

biomedical events. The HANN [36] model relies on the GCN and hypergraph to obtain local and global contexts. The KB-driven tree LSTM [33] depends on KB concept embedding to improve the pretrained distributed word representations. The Graph Edge-conditioned Attention Networks with Science BERT (GEANet-SciBERT) [34] adopts a hierarchical graph representation encoded by graph edge-conditioned attention networks to incorporate domain knowledge from the Unified Medical Language System into a pretrained language model. [Table 7](#) illustrates that except for the DeepEventMine, our approach outperformed all previous methods.

Table 7. The performance of biomedical event extraction on the BioNLP shared task 2011 Genia event corpus.

Method and event type	Precision (%)	Recall (%)	F ₁ score (%)
TEES^{a,b}			
Event total ^c	57.65	49.56	53.30
EventMine^a			
Event total	63.48	53.35	57.98
Stacked generalization^a			
Event total	66.46	48.96	56.38
TEES-CNNs^{a,d}			
Event total	69.45	49.94	58.07
HANN^{e,f}			
Event total	71.73	53.21	61.10
KB^g-driven tree LSTM^{e,h}			
Simple total ⁱ	85.95	72.62	78.73
Binding	53.16	37.68	44.10
Regulation total ^j	55.73	41.73	47.72
Event total	67.10	52.14	58.65
GEANet-SciBERT^{e,k}			
Regulation total	55.21	47.23	50.91
Event total	64.61	56.11	60.06
DeepEventMine^e			
Regulation total	62.36	51.88	56.64 ^l
Event total	76.28	55.06	63.96 ^l
Our model^e			
Simple total	82.23	78.88	80.52
Binding	55.12	37.48	44.62
Regulation total	57.82	46.39	51.48
Event total	72.62	53.33	61.50

^aPipeline model.

^bTEES: Turku Event Extraction System.

^cRepresents the overall performance on the test set.

^dCNN: convolutional neural network.

^eJoint model.

^fHANN: hierarchical artificial neural network.

^gKB: knowledge base.

^hLSTM: long short-term memory.

ⁱRepresents the overall performance for simple events on the test set.

^jRepresents the overall performance for nested events on the test set (including regulation, positive regulation, and negative regulation subevents).

^kGEANet-SciBERT: Graph Edge-conditioned Attention Networks with Science BERT.

^lThe best value compared with other models.

The KB-driven tree LSTM and GEANet-SciBERT both draw on the KB to enhance the semantic representation of words to improve the extraction performance of nested (regulation) events. However, the KB-driven tree LSTM only leverages

traditional static word embedding, which cannot deeply integrate information from the KB; thus, its performance on nested events is unsatisfactory.

Unlike the KB-driven tree LSTM method, the GEANet-SciBERT model uses a specialized medical KB and scientific information to enrich the dynamic semantic representation of Bidirectional Encoder Representation from Transformers (BERT) and enhances the capability of inferring nested events via a novel GNN. Thus, the F_1 scores for the nested event extraction were significantly boosted.

Interestingly, the DeepEventMine had an outstanding performance for extracting nested biomedical events on BioNLP-ST 2011 GE but had a passive performance on MLEE. There are three reasons for this fact. First, the DeepEventMine model jointly learns 4 biomedical information tasks (entity detection, trigger detection, role detection, and event detection), which can share more biomedical features and knowledge when model training. Second, the DeepEventMine model uses a more complex graph structure (multiple overlapping directed acyclic graphs) to obtain rich syntactic information. (Finally, the BioNLP-ST 2011 GE data set size is larger than that of the MLEE data set; thus, the DeepEventMine model can be fully trained on a large corpus and enhance the performance of extracting nested events.

Discussion

In this section, we will study and discuss the performance of our CPJE model using the MLEE corpus.

Ablation Study

The Impact of the BiLSTM

Although the output of BioBERT contains rich semantic information, it has some noise impact on semantic information after concatenating POS embedding, entity embedding, and BioBERT embedding. In addition, the dimension of the BioBERT output is 768, and the total size after concatenation is more extensive, which tends to cause the phenomenon of combination explosion in the feature space. Therefore, we considered using a BiLSTM, which reduces the total dimension and integrates other information with the BioBERT information to obtain a richer semantic representation.

If we remove the BiLSTM layer, the trigger recognition precision is dropped from 82.20% to 75.64%, and the trigger recognition F_1 score is dropped from 80.18% to 76.39%, which further affects the event extraction performance (the event extraction F_1 score is fell from 62.80% to 58.02%).

The Impact of Softmax Probability

To evaluate the contribution of the softmax probability distribution after trigger prediction to the event extraction task, we used the traditional joint extraction method (as shown in equation 10), which only uses source information when extracting candidate trigger vectors and event argument vectors.

If we only use the source information (soft trigger) for joint extraction, the event extraction task lacks the probability

distribution information after trigger recognition, which results in a decline in the recall rate of the model and further affects the F_1 scores (the event extraction F_1 score is dropped from 62.80% to 60.09%). However, the overall result is still slightly higher than the pipeline baseline, which also reflects that joint extraction can eliminate cascading errors.

The Impact of GCN

We removed the syntactic structure to evaluate the importance of the GCN network; therefore, the GCN module was useless in our model. If the model lacks the GCN component, the performance of trigger recognition is slightly degraded (the trigger recognition F_1 score is fell from 80.18% to 78.78%), and the result of event extraction is significantly worse than that of the proposed model (the event extraction F_1 score is fell from 62.80% to 58.40%).

As the syntactic structure can provide significant potential information for event extraction, the GCN model can be aware of the direction of information flow in syntactic structures and capture these features effectively. Therefore, the GCN model is vital for event extraction.

The Impact of Dice Loss

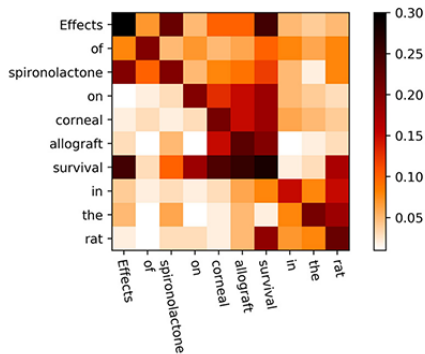
In the face of an imbalance in biomedical corpora, we used the Dice loss function. To verify that the Dice loss function had a better effect on event extraction, we used the cross-entropy loss function for comparison.

A significantly large number of negative examples in the data set indicates that easy-negative examples are extensive. A large number of straightforward examples overwhelmed the training, making the model insufficient to distinguish between positive and hard-negative examples. As the cross-entropy loss is accuracy oriented and each instance contributes equally to the loss function, the precision of the model increases (the event extraction precision is risen from 72.26% to 89.26%), but the F_1 scores do not increase (the event extraction F_1 score is dropped from 62.60% to 60.30%). Dice loss is a muted version of the F_1 score—the harmonic mean of precision and recall. When the positive and negative examples in the data set are unbalanced, the Dice loss will reduce the focus on the easy-negative sample and increase the attention on positive and hard-negative samples, thereby balancing the precision and recall values and increasing the F_1 scores.

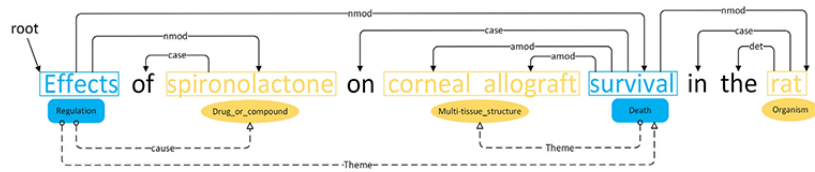
Visualization

For the effectiveness of the attention-based gate GCN, we used the sentence “Effects of spironolactone on corneal allograft survival in the rat” in Figure 3 as an example to illustrate the captured interaction features. From Figure 3B, we know this sentence contains 2 events: a *regulation* event caused by *effects* and a *death* event caused by *survival*. In addition, a death event is one of the arguments for the regulation event.

Figure 3. An example of attention-based gate graph neural network effectiveness. (A) Row-wise heap map, where each row is an array of average scores of the 2 heads obtained from the multi-head attention mechanism. The darker the color, the higher the score and the stronger the interaction. (B) Dependency parsing result produced by Stanford CoreNLP and the golden relationships between event triggers and arguments, where yellow boxes represent entity type, and the blue boxes represent event type.



(A) Visualization of the sentence attention scores



(B) Dependency parsing result and event extraction result

As we can see in Figure 3A, the *effects* row has moderately strong links with *Effects* (self), spironolactone (its argument), and *survival* (its argument and another event). Meanwhile, the *survival* row has strong links with *survival* (self), *effects* (another event), and *corneal allograft* (its argument). In addition, the words *rat* and *on* also have strong connections with *survival*, which means that the syntactic dependency information generated by parsing is propagated through the GCN.

Case Study

Overview

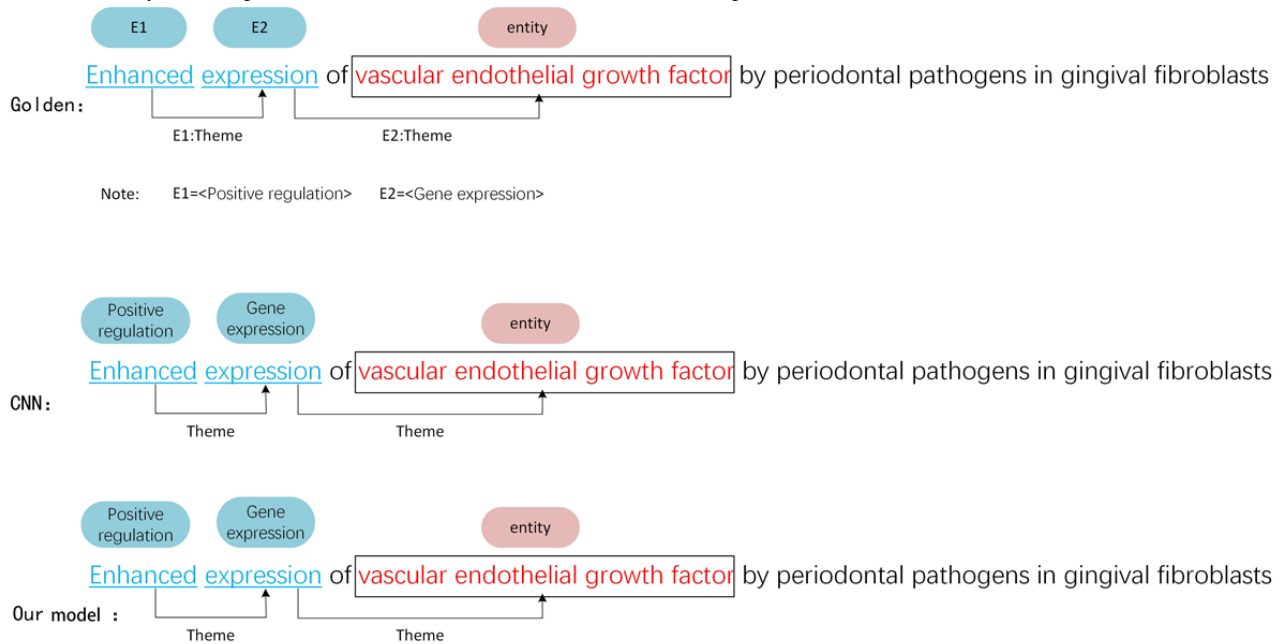
Our framework has not achieved state-of-the-art results for the BioNLP-ST 2011 GE corpus. However, the performance of extracting nested biomedical events is satisfactory, particularly in the MLEE corpus. To more intuitively demonstrate the

performance of our model in extracting nested biomedical events, we analyzed 3 examples of nested events selected from the MLEE test set to study the strengths and weaknesses of our model compared with the CNN [3].

Case 1

As shown in Figure 4, case 1 is a simple nested event, where the role type of event argument is only the *theme*. It is a nested event; however, both the CNN and our model obtained correct event extraction results. This is because this sentence does not have a complete component, and perhaps, it is only a part of a complete sentence. The simpler the sentence structure is, the easier it is for the model to extract practical features. Therefore, the extraction performance for such nested events is generally favorable.

Figure 4. Case study for a simple nested event on the multilevel event extraction corpus. CNN: convolutional neural network.

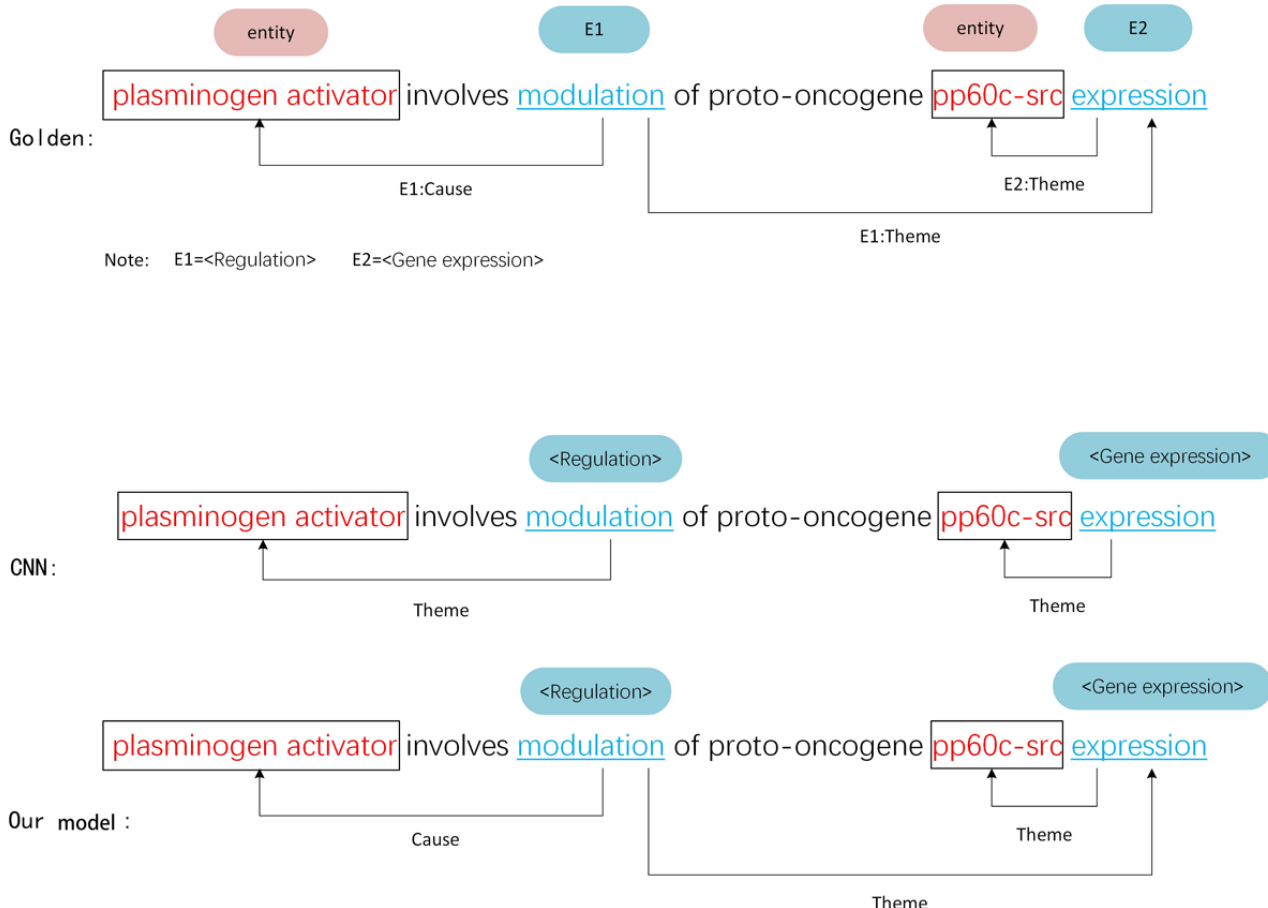


Case 2

Case 2 is a general nested event whose sentence component is complete, and the role types of event arguments are *theme* and *cause*. As shown in Figure 5, the CNN model detects all correct event triggers but cannot detect the correct event arguments. The CNN model is a pipeline approach that considers trigger recognition and argument detection tasks in a cascade rather than a parallel relationship. In general, they first input the text into the CNN model to identify the triggers in the sentence. Then, they construct <trigger, entity> or <trigger, trigger>

candidate pairs and input them into the CNN model again to detect the arguments. Finally, rule-based or machine learning-based methods are used to postprocess triggers and arguments to construct complete biomedical events. If there is an error in some of these steps, it will directly affect the performance of event extraction. However, our joint method regards trigger recognition and argument detection as parallel tasks that can provide valid information. Thus, we trained both tasks jointly with one model, and errors could only be generated during the model training.

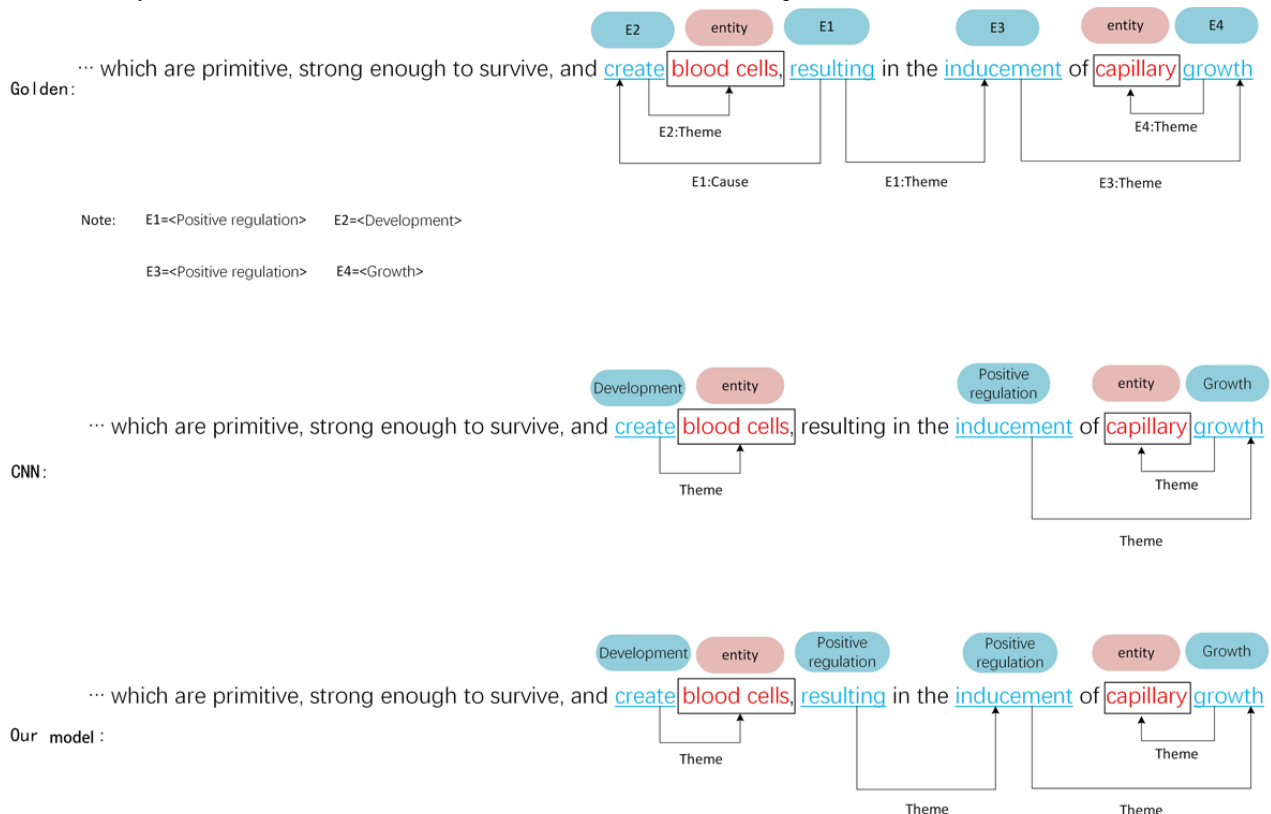
Figure 5. Case study for a common nested event on multilevel event extraction corpus. CNN: convolutional neural network.



Case 3

Case 3 is a cross-sentence nested event, as shown in Figure 6. From this example, we can determine what needs to be improved. As multiple events are nested in each other, and some of these events are not in the same sentence, this prevents the

model from extracting all events efficiently and accurately. Compared with the CNN model, although our model can identify the *positive regulation* event triggered by *resulting*, it is not in the same clause as the *development* event triggered by *create*, which causes the *positive regulation* event to lack an event argument.

Figure 6. Case study for an across-sentence nested event on multilevel event extraction corpus. CNN: convolutional neural network.

Conclusions

In this study, a CPJE framework based on a multi-head attention graph CNN is proposed to achieve biomedical event extraction tasks. The cascading errors between the 2 subtasks were reduced because of the use of the joint extraction framework. With the help of the attention-based gate GCN, syntactic dependency information and the interrelations between triggers and related entities were effectively learned; thus, the extraction

performance of nested biomedical events improved. The Dice loss replaced the cross-entropy loss, which weakened the negative impact of the imbalanced data set. Overall, the model obtained the best F_1 score in the MLEE biomedical event extraction corpus and achieved favorable performance on the BioNLP-ST 2011 GE corpus. In the future, we will consider integrating external resource knowledge to allow the model to learn richer information and improve the performance of cross-sentence nested events.

Acknowledgments

This study was funded by grants from the National Natural Science Foundation of China (number 62072070).

Authors' Contributions

YW proposed the study of biomedical event extraction, implemented and verified the effectiveness of the joint extraction framework, and wrote the first draft. JW put forward constructive suggestions for revising this draft. H Lu read the final manuscript and provided some useful suggestions. H Lin read and approved the final manuscript. BX read and approved the final manuscript. YZ helped to review and revise the draft. SKB helped revise the draft.

Conflicts of Interest

None declared.

References

- McDonald RT, Pereira FC, Kulick SN, Winters R, Jin Y, White PS. Simple algorithms for complex relation extraction with applications to biomedical IE. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics. 2005 Presented at: ACL '05; June 25-30, 2005; Ann Arbor, MI, USA p. 491-498. [doi: [10.3115/1219840.1219901](https://doi.org/10.3115/1219840.1219901)]
- Kilicoglu H, Bergler S. Effective bio-event extraction using trigger words and syntactic dependencies. Comput Intell 2011 Nov 27;27(4):583-609. [doi: [10.1111/j.1467-8640.2011.00401.x](https://doi.org/10.1111/j.1467-8640.2011.00401.x)]

3. Wang A, Wang J, Lin H, Zhang J, Yang Z, Xu K. A multiple distributed representation method based on neural network for biomedical event extraction. *BMC Med Inform Decis Mak* 2017 Dec 20;17(Suppl 3):171 [FREE Full text] [doi: [10.1186/s12911-017-0563-9](https://doi.org/10.1186/s12911-017-0563-9)] [Medline: [29297321](https://pubmed.ncbi.nlm.nih.gov/29297321/)]
4. Björne J, Salakoski T. Biomedical event extraction using convolutional neural networks and dependency parsing. In: *Proceedings of the BioNLP 2018 workshop*. 2018 Presented at: BioNLP '18; July 19, 2018; Melbourne, Australia p. 98-108. [doi: [10.18653/v1/w18-2311](https://doi.org/10.18653/v1/w18-2311)]
5. He X, Li L, Song X, Huang D, Ren F. Multi-level attention based BLSTM neural network for biomedical event extraction. *IEICE Trans Inf Syst* 2019;E102.D(9):1842-1850. [doi: [10.1587/transinf.2018edp7268](https://doi.org/10.1587/transinf.2018edp7268)]
6. Pyysalo S, Ohta T, Miwa M, Cho H, Tsujii J, Ananiadou S. Event extraction across multiple levels of biological organization. *Bioinformatics* 2012 Sep 15;28(18):i575-i581 [FREE Full text] [doi: [10.1093/bioinformatics/bts407](https://doi.org/10.1093/bioinformatics/bts407)] [Medline: [22962484](https://pubmed.ncbi.nlm.nih.gov/22962484/)]
7. Kim JD, Wang Y, Takagi T, Yonezawa A. Overview of Genia event task in BioNLP shared task 2011. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. 2011 Presented at: BioNLP '11; June 24, 2011; Portland, OR, USA p. 7-15. [doi: [10.1186/1471-2105-13-s11-s1](https://doi.org/10.1186/1471-2105-13-s11-s1)]
8. Zheng S, Hao Y, Lu D, Bao H, Xu J, Hao H, et al. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* 2017 Sep 27;257:59-66. [doi: [10.1016/j.neucom.2016.12.075](https://doi.org/10.1016/j.neucom.2016.12.075)]
9. Ye ZX, Ling ZH. Distant supervision relation extraction with intra-bag and inter-bag attentions. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019 Presented at: NAACL '19; June 2-7, 2019; Minneapolis, MN, USA p. 2810-2819. [doi: [10.48550/arXiv.1904.00143](https://doi.org/10.48550/arXiv.1904.00143)]
10. Feng J, Huang M, Zhao L, Yang Y, Zhu X. Reinforcement learning for relation classification from noisy data. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2018 Feb Presented at: AAAI '18; Feb 2-7, 2018; New Orleans, LA, USA.
11. Fu TJ, Li PH, Ma WY. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019 Presented at: ACL '19; July 28-August 2, 2019; Florence, Italy p. 1409-1418. [doi: [10.18653/v1/p19-1136](https://doi.org/10.18653/v1/p19-1136)]
12. Guo Z, Zhang Y, Lu W. Attention guided graph convolutional networks for relation extraction. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019 Presented at: ACL '19; July 28-August 2, 2019; Florence, Italy p. 241-251. [doi: [10.18653/v1/p19-1024](https://doi.org/10.18653/v1/p19-1024)]
13. Li Q, Ji H, Huang L. Joint event extraction via structured prediction with global features. In: *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. 2013 Aug Presented at: ACL '13; August 4-9, 2013; Sofia, Bulgaria p. 73-82.
14. Keith KA, Handler A, Pinkham M, Magliozzi C, McDuffie J, O'Connor B. Identifying civilians killed by police with distantly supervised entity-event extraction. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017 Sep Presented at: EMNLP '17; September 7-8, 2017; Copenhagen, Denmark p. 1547-1557. [doi: [10.18653/v1/d17-1163](https://doi.org/10.18653/v1/d17-1163)]
15. Reichart R, Barzilay R. Multi-event extraction guided by global constraints. In: *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2012 Jun Presented at: NAACL '12; June 3-8, 2012; Montreal, Canada p. 70-79.
16. Lu W, Roth D. Automatic event extraction with structured preference modeling. In: *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*. 2012 Jul Presented at: ACL '12; July 8-14, 2012; Jeju Island, Korea p. 835-844.
17. Sha L, Qian F, Chang B, Sui Z. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. 2018 Presented at: AAAI '18; February 2-7, 2018; New Orleans, LA, USA.
18. Nguyen TH, Grishman R. Modeling skip-grams for event detection with convolutional neural networks. In: *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. 2016 Presented at: EMNLP '16; November 1-5, 2016; Austin, TX, USA p. 886-891. [doi: [10.18653/v1/d16-1085](https://doi.org/10.18653/v1/d16-1085)]
19. Liu X, Luo Z, Huang H. Jointly multiple events extraction via attention-based graph information aggregation. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018 Presented at: EMNLP '18; October 31-November 4, 2018; Brussels, Belgium p. 1247-1256. [doi: [10.18653/v1/d18-1156](https://doi.org/10.18653/v1/d18-1156)]
20. Kim JD, Ohta T, Pyysalo S, Kano Y, Tsujii J. Overview of BioNLP'09 shared task on event extraction. In: *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*. 2009 Presented at: BioNLP '09; June 05, 2009; Boulder, CO, USA p. 1-9. [doi: [10.3115/1572340.1572342](https://doi.org/10.3115/1572340.1572342)]
21. Bossy R, Golik W, Ratkovic Z, Bessières P, Nédellec C. Bionlp shared task 2013 - an overview of the bacteria biotope task. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013 Presented at: BioNLP '13; August 09, 2013; Sofia, Bulgaria p. 161-169. [doi: [10.18653/v1/W16-3002](https://doi.org/10.18653/v1/W16-3002)]
22. Miwa M, Saetre R, Kim JD, Tsujii J. Event extraction with complex event classification using rich features. *J Bioinform Comput Biol* 2010 Feb;8(1):131-146. [doi: [10.1142/s0219720010004586](https://doi.org/10.1142/s0219720010004586)] [Medline: [20183879](https://pubmed.ncbi.nlm.nih.gov/20183879/)]

23. Miwa M, Thompson P, Ananiadou S. Boosting automatic event extraction from the literature using domain adaptation and coreference resolution. *Bioinformatics* 2012 Jul 01;28(13):1759-1765 [[FREE Full text](#)] [doi: [10.1093/bioinformatics/bts237](https://doi.org/10.1093/bioinformatics/bts237)] [Medline: [22539668](https://pubmed.ncbi.nlm.nih.gov/22539668/)]
24. Björne J, Salakoski T. TEES 2.1: automated annotation scheme learning in the BioNLP 2013 shared task. In: *Proceedings of the BioNLP Shared Task 2013 Workshop*. 2013 Presented at: BioNLP '13; August 9, 2013; Sofia, Bulgaria p. 16-25. [doi: [10.18653/v1/w16-3009](https://doi.org/10.18653/v1/w16-3009)]
25. Majumder A, Ekbal A, Naskar SK. Biomolecular event extraction using a stacked generalization-based classifier. In: *Proceedings of the 13th International Conference on Natural Language Processing*. 2016 Presented at: ICNLP '16; December 17-20, 2016; Varanasi, India p. 55-64.
26. Riedel S, McCallum A. Robust biomedical event extraction with dual decomposition and minimal domain adaptation. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. 2011 Presented at: BioNLP '11; June 24, 2011; Portland, OR, USA p. 46-50.
27. Venugopal D, Chen C, Gogate V, Ng V. Relieving the Computational Bottleneck: joint inference for event extraction with high-dimensional features. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. 2014 Presented at: EMNLP '14; October 25-29, 2014; Doha, Qatar p. 831-843. [doi: [10.3115/v1/d14-1090](https://doi.org/10.3115/v1/d14-1090)]
28. Nguyen DQ, Verspoor K. From POS tagging to dependency parsing for biomedical event extraction. *BMC Bioinformatics* 2019 Feb 12;20(1):72 [[FREE Full text](#)] [doi: [10.1186/s12859-019-2604-0](https://doi.org/10.1186/s12859-019-2604-0)] [Medline: [30755172](https://pubmed.ncbi.nlm.nih.gov/30755172/)]
29. Zhou D, Zhong D. A semi-supervised learning framework for biomedical event extraction based on hidden topics. *Artif Intell Med* 2015 May;64(1):51-58. [doi: [10.1016/j.artmed.2015.03.004](https://doi.org/10.1016/j.artmed.2015.03.004)] [Medline: [25863986](https://pubmed.ncbi.nlm.nih.gov/25863986/)]
30. Rao S, Marcu D, Knight K, Daumé III H. Biomedical event extraction using abstract meaning representation. In: *Proceedings of the BioNLP 2017 workshop*. 2017 Presented at: BioNLP '17; August 04, 2017; Vancouver, Canada p. 126-135. [doi: [10.18653/v1/w17-2315](https://doi.org/10.18653/v1/w17-2315)]
31. Yan S, Wong KC. Context awareness and embedding for biomedical event extraction. *Bioinformatics* 2020 Jan 15;36(2):637-643. [doi: [10.1093/bioinformatics/btz607](https://doi.org/10.1093/bioinformatics/btz607)] [Medline: [31392318](https://pubmed.ncbi.nlm.nih.gov/31392318/)]
32. Zhao W, Zhao Y, Jiang X, He T, Liu F, Li N. A novel method for multiple biomedical events extraction with reinforcement learning and knowledge bases. In: *Proceedings of the 2020 IEEE International Conference on Bioinformatics and Biomedicine*. 2020 Presented at: BIBM '20; December 16-19, 2020; Seoul, South Korea p. 402-407. [doi: [10.1109/bibm49941.2020.9313214](https://doi.org/10.1109/bibm49941.2020.9313214)]
33. Li D, Huang L, Ji H, Han J. 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*. 2019 Presented at: NAACL '19; June 2-7, 2019; Minneapolis, MN, USA p. 1421-1430. [doi: [10.18653/v1/N19-1145](https://doi.org/10.18653/v1/N19-1145)]
34. Huang KH, Yang M, Peng N. Biomedical event extraction with hierarchical knowledge graphs. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2020 Presented at: EMNLP '20; November 16-20, 2020; Virtual p. 1277-1285. [doi: [10.18653/v1/2020.findings-emnlp.114](https://doi.org/10.18653/v1/2020.findings-emnlp.114)]
35. Trieu HL, Tran TT, Duong KN, Nguyen A, Miwa M, Ananiadou S. DeepEventMine: end-to-end neural nested event extraction from biomedical texts. *Bioinformatics* 2020 Dec 08;36(19):4910-4917 [[FREE Full text](#)] [doi: [10.1093/bioinformatics/btaa540](https://doi.org/10.1093/bioinformatics/btaa540)] [Medline: [33141147](https://pubmed.ncbi.nlm.nih.gov/33141147/)]
36. Zhao W, Zhang J, Yang J, He T, Ma H, Li Z. A novel joint biomedical event extraction framework via two-level modeling of documents. *Inf Sci* 2021 Mar;550:27-40. [doi: [10.1016/j.ins.2020.10.047](https://doi.org/10.1016/j.ins.2020.10.047)]
37. Ramponi A, van der Goot R, Lombardo R, Plank B. Biomedical event extraction as sequence labeling. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*. 2020 Presented at: EMNLP '20; November 16-20, 2020; Virtual p. 5357-5367. [doi: [10.18653/v1/2020.emnlp-main.431](https://doi.org/10.18653/v1/2020.emnlp-main.431)]
38. Lee J, Yoon W, Kim S, Kim D, Kim S, So CH, et al. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 2020 Feb 15;36(4):1234-1240 [[FREE Full text](#)] [doi: [10.1093/bioinformatics/btz682](https://doi.org/10.1093/bioinformatics/btz682)] [Medline: [31501885](https://pubmed.ncbi.nlm.nih.gov/31501885/)]
39. Klein D, Manning CD. Accurate unlexicalized parsing. In: *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*. 2003 Presented at: ACL '03; July 7-12, 2003; Sapporo, Japan p. 423-430. [doi: [10.3115/1075096.1075150](https://doi.org/10.3115/1075096.1075150)]
40. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, et al. Attention is all you need. In: *Proceedings of Annual Conference on Advances in Neural Information Processing Systems*. 2017 Presented at: NIPS '17; December 4-9, 2017; Long Beach, CA, USA.
41. Li X, Sun X, Meng Y, Liang J, Wu F, Li J. Dice loss for data-imbalanced NLP tasks. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020 Presented at: ACL '20; July 5-10, 2020; Virtual p. 465-476. [doi: [10.18653/v1/2020.acl-main.45](https://doi.org/10.18653/v1/2020.acl-main.45)]
42. Björne J, Salakoski T. Generalizing biomedical event extraction. In: *Proceedings of BioNLP Shared Task 2011 Workshop*. 2011 Presented at: BioNLP '11; June 24, 2011; Portland, OR, USA p. 183-191.

Abbreviations

BERT: Bidirectional Encoder Representation From Transformers
BiLSTM: bidirectional long short-term memory
BioBERT: Biomedical Bidirectional Encoder Representation From Transformers
BioNLP: biomedical natural language processing
BioNLP-ST: biomedical natural language processing shared task
CNN: convolutional neural network
CPJE: conditional probability joint extraction
GCN: graph convolutional network
GE: Genia event
GEANet-SciBERT: Graph Edge-conditioned Attention Networks with Science BERT
GNN: graph neural network
HANN: hierarchical artificial neural network
KB: knowledge base
LSTM: long short-term memory
mdBLSTM: bidirectional long short-term memory with a multilevel attention mechanism and dependency-based word embeddings
MLEE: multilevel event extraction
POS: parts of speech
RL: reinforcement learning
SGD: stochastic gradient descent
TEES: Turku Event Extraction System

Edited by T Hao; submitted 08.03.22; peer-reviewed by T Zhang, Y An; comments to author 06.04.22; revised version received 15.04.22; accepted 19.04.22; published 07.06.22

Please cite as:

Wang Y, Wang J, Lu H, Xu B, Zhang Y, Banbhrani SK, Lin H

Conditional Probability Joint Extraction of Nested Biomedical Events: Design of a Unified Extraction Framework Based on Neural Networks

JMIR Med Inform 2022;10(6):e37804

URL: <https://medinform.jmir.org/2022/6/e37804>

doi: [10.2196/37804](https://doi.org/10.2196/37804)

PMID: [35671070](https://pubmed.ncbi.nlm.nih.gov/35671070/)

©Yan Wang, Jian Wang, Huiyi Lu, Bing Xu, Yijia Zhang, Santosh Kumar Banbhrani, Hongfei Lin. Originally published in JMIR Medical Informatics (<https://medinform.jmir.org>), 07.06.2022. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIR Medical Informatics, is properly cited. The complete bibliographic information, a link to the original publication on <https://medinform.jmir.org/>, as well as this copyright and license information must be included.