COMPLEX EVENT EXTRACTION ALGORITHM BASED ON DEEP EMBEDDING DECOUPLING

Hong Zhou LIAO[1]

Complex events are descriptions of events with multiple subjects, objects and behaviors. Compound sentence is the most typical representative, complex events have problems such as sentence nesting, element dispersion and sharing. As a result, the conventional event element extraction algorithm cannot completely extract the corresponding nested event elements. To deal with the above problem, this paper proposes a multi-domain oriented complex event extraction algorithm based on deep embedded decoupling (CEEN-DEP). First, build Roberta-Bigru-Attention network to fuse character features, time series features, interaction features and embed semantic features of long text complex events. Then being aimed at nested attributes for complex events, define a common tag system for multi-domain complex event elements and divide complex event elements into common elements and domain elements. At the same time, considering the dependence between coupling tags, the conditional random field is introduced, and feedback training is conducted to complete the joint extraction of coupling elements. The comparative experiments show that the algorithm can decouple the nested elements of complex events and complete the reorganization of corresponding event elements. The accuracy rate, recall rate and f1 value are relatively stable under different sample sizes, and they are maintained in the range of 80%-95%, which proves the effectiveness and robustness of the algorithm.

Keywords: complex events; element extraction; deep embedded decoupling; tag system

1. Introduction

Event element extraction is an important but challenging task in information extraction. As a special form of information, an event occurs at a specific time and place, involving one or more participants, and can usually be described as a change in state. The event extraction task aims to extract such event information from unstructured plain text into a structured form, mainly describing when, where, why, and how events occurred in the real world, as well as what specifically happened, and who participated in them [1]. In application, event extraction facilitates people to retrieve event information and analyze their behavior and is often used as an upstream task for information retrieval, recommendation, intelligent question answering, knowledge map construction, and other applications. Among them, complex event element extraction often

[1] Second Key Laboratory, Southwest Electronic Technology Research Institute, China, e-mail: liao_hongzhou@163.com

leads to problems such as sentence nesting and element dispersion in its events, which often leads to the inability of general sentence level event element extraction algorithms to extract corresponding event elements completely, and this problem cannot be avoided in different fields [2]. Starting from the essence of the problem that complex event elements are difficult to extract, this article takes a typical compound sentence as an example to classify and jointly extract complex event elements, which can avoid error transmission in different stages of pipeline element extraction and can also achieve one-to-one correspondence of event elements at different locations, forming a clear and complete complex event element. At the same time, the method has less domain limitations and strong robustness, and it can be applied to multiple different fields.

As a hot research direction in the field of natural language processing in recent years, event element extraction has been receiving continuous attention from scholars at home and abroad. Relevant research has also experienced a transformation process from early template matching methods to medium-term machine learning methods, and then to the current deep learning methods. Due to the limitations of data and computational power, early template matching is a mainstream method for event element extraction. For example, Kim et al. [3] used language models for event element information extraction, Gupta et al. [4] proposed a template-based approach for news event element matching, Kitani et al. [5] designed and developed a Japanese information extraction system that merges information using a pattern matcher and discourse processor. Wang et al. [6] proposed a novel approach to extract event semantic elements, which employs semantic role labeling (SRL) enhanced by heuristic rules to extract event 5W1H (Who, What, Whom, When, Where and How) elements. Template matching methods can achieve good performance in specific fields through expert experience guidance, but their portability is not strong. In the medium term, machine learning algorithms are mainly based on special engineering methods. Compared to template matching algorithms, they have been transformed from expert driven to data-expert driven, reducing dependence on domain experts, and significantly enhancing portability. For example, Ahn et al. [7] took the lead in using lexical features, dictionary features, syntactic features, and other subtasks to extract elements into four stages in 2006, while Li et al. [8] extracted all event information simultaneously from global features and overall structures in 2014. Compared to the previous two methods, deep learning algorithms have stronger data-driven capabilities. In the current context of big data and high computational power, the performance of event element extraction has been significantly improved. Therefore, as deep learning continues to expand in the field of natural language processing, it has become the mainstream method of event element extraction at present. For example, with the success of BERT, the pre training language model has also been used in event extraction. Based on the pre training

language model, Wadden et al. [9] studied how to better utilize rich event knowledge in large-scale unsupervised data to improve performance. Based on the BERT pre-trained model, Portelli et al. [10] proposed for the first time the use of the SpanBERT architecture for the task of ADE extraction and new version of the popular BERT transformer showed improved capabilities with multi-token text spans. Si et al. [11] introduced a learning strategy based on Prompt into the field of event extraction for the first time, automatically utilizing tag semantics at the input and output ends, reducing practical requirements such as hardware and data, while improving the accuracy of element extraction. Ma et al. [12] tended to model event extraction as a classification task and Jiang et al. [13] solved the event extraction in sequence labeling manner by tagging the sentence only once, which may not solve the overlapping problem. Du et al. [14] used a common structure to uniformly model various tasks, including event detection and event argument extraction and the output structure can be a sentence filled with slotted templates, at the same time, Lu et al. [15] proposed that the output structure can be a sentence filled with some linearly serialized tree structure. Liu et al. [16] thought that the model may pay more attention to global features and ignore local details. Based on that, Liu et al. used prompt-based approaches to force the model to focus on specific pieces of information to control its output for different event types.

The current research on event extraction mainly focuses on sentence level extraction, namely, simple sentence extraction. Compared to simple sentences, complex sentence extraction has problems such as element dispersion and argument sharing, which has gradually become a hot and difficult topic in current research. Some scholars have begun to make breakthroughs. For example, Wang et al. [17] proposed a relation-aware Transformer-based Document-level Joint Event Extraction model (TDJEE), which encodes relations between words into the context and leverages modified Transformer to capture document-level information to fill event arguments. Tong et al. [18] presented DocEE, a new document-level event extraction dataset including 27,000+ events, 180,000+ arguments, but experiments indicated that DocEE is an open issue. Shun et al. [19] proposed a novel end-to-end financial document element extraction model, Doc2EDAG, Entity based directed acyclic graphs can be generated to achieve event extraction at the financial document level. The current research on complex event extraction mainly focuses on document level central idea element extraction, which is based on text level element coding to complete core element extraction and complement, while focusing more on the financial field. Based on the current mature research on simple sentence extraction and document level element extraction, this article focuses on the research on complex event extraction between the two, namely, compound sentence element extraction,

which can further expand simple sentence extraction, promote document level element extraction, and improve text comprehension.

2. Methods

This paper proposes a multi-domain oriented complex event extraction algorithm based on deep embedded decoupling (CEEN-DEP), which integrates pre-trained language models, gated recurrent unit, and self-attention mechanisms to achieve context fusion at the text level and complete complex event element extraction for long text in multiple domains. First, the proposed algorithm in the paper receives sentence fragments, and each sentence fragment is encoded into low-dimensional dense vectors by pre-trained language model (Roberta). Then, the paper proposes to splice the sentence fragment vectors according to the sequence of sentence fragments and use recurrent neural network model (Bigru) to capture temporal features of different sentence fragment vectors. Next, using attention mechanism to form higher-level semantic vectors by finding the importance and relationship between low-dimensional temporal vectors. Last, higher-level semantic vectors are used as an observation sequence to predict their most likely marker sequence, namely the following universal complex event tag proposed in this paper, by graphical model (CRF). The overall framework is shown in Fig. 1.

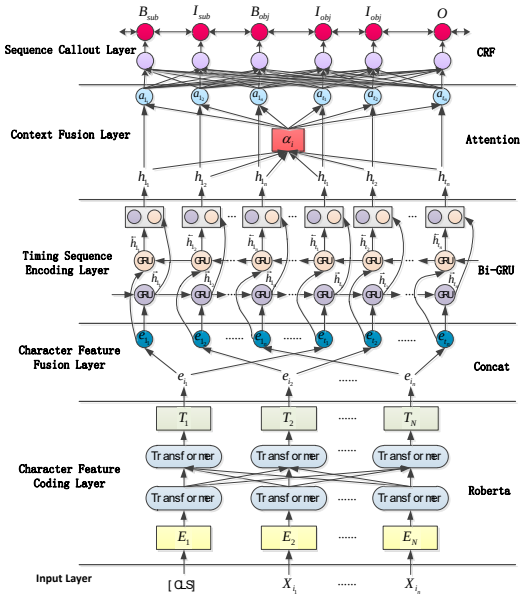

Fig. 1. Framework of complex event element extraction model

2.1 Character feature encoding

The character feature encoding layer performs word embedding encoding using the Roberta pre-trained model [20] to obtain dense word vector features.

The Roberta model evolved from the Bert model [21], and its main structure includes multi-heads self-attention and feedforward neural networks. The structure of the word feature encoder is shown in Fig. 2.

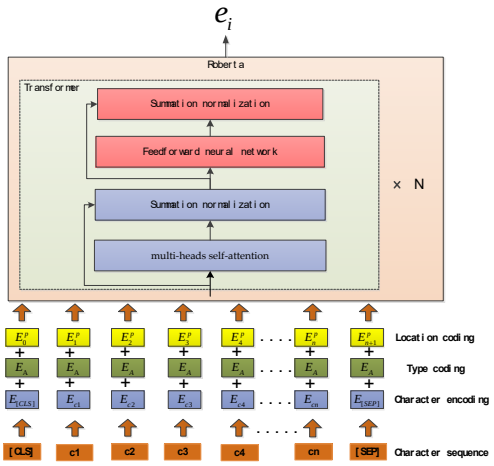

Fig. 2. Character feature coding based on Roberta

Where, $e_i = Roberta(E_c + E_A + E^p)$, e_i represents the character feature encoding tensor, i represents the sample, $0 < i < n$, n represents the maximum number of samples, E_c represents the character encoding matrix, E_A represents the type encoding matrix, E^p represents the position encoding matrix, and $e_i, E_c, E_A, E^p \in R^{m \times d}$, m represents the maximum length of the input sequence, and d represents the character encoding dimension.

Compared to the mature Bert model, character feature encoding based on Roberta is mainly optimized in terms of training methods. By changing static encoding to dynamic encoding, the same sample has different mask positions each time, which can learn more semantic representations and improve the quality of context association. At the same time, next sentence prediction task based on Bert is discarded, which can reduce training costs.

2.2 Character feature fusion

The encoding tensor of a sentence sequence can be obtained through character encoding based on Roberta, but the pre-trained Roberta is limited in coding length, and it is unable to represent and learn complex events in long texts. Therefore, this article first adopts the "concat" method to splice the encoded values of multiple sentence segments to achieve the fusion of character features, facilitating subsequent sequential encoding and learning of character features. The mathematical expression is shown in equation:

$$e = concat(e_t) \tag{1}$$

where, e_t represents the character feature tensor of the t-th sentence segment sequence in the character feature coding layer, $e_t \in R^{m \times d}$, e represents the

output tensor of the character feature fusion layer, $e \in R^{l \times d}$ and $l = m \times k$, l represents the sequence length after character feature fusion, m represents the maximum length of the input sequence, d represents the character encoding dimension, and k represents the maximum number of sentence fragments.

2.3 Timing sequence encoding

Character feature fusion can expand the length of input sequences and obtain feature representations of complex sentences in long texts, but it does not take into account the positional relationship and sequence characteristics between sentence segments. Therefore, this paper uses a bidirectional gated recurrent unit (Bi-GRU) for timing coding to obtain sequence characteristics between sentence segments. Bi-GRU is an iterative evolution of GRU networks [22], which can better encode timing characteristics through positive and negative bidirectional coding. The GRU unit mainly includes a reset gate and an update gate, and the mathematical expressions are shown in Equation (2) and Equation (3) respectively:

$$r_t = sigmoid(U_r x_t + W_r h_{t-1} + b_r) \tag{2}$$

$$z_t = sigmoid(U_z x_t + W_z h_{t-1} + b_z) \tag{3}$$

where, x_t represents the input of the t-th GRU unit, that is, the output tensor of the t-th position of the character feature fusion layer, h_{t-1} represents the output of the previous GRU unit, U_r, W_r, b_r represents the weight parameters and offsets that need to be learned by the reset gate, and U_z, W_z, b_z represents the weight parameters and offsets that need to be learned by the update gate.

Next, calculate the forward hidden layer output as follows:

$$\vec{h}_t = z_t \times h_{t-1} + (1 - z_t) \times \tilde{h}_t \tag{4}$$

In Equation (4), indicating the candidate hidden layer status can be expressed as

$$\tilde{h}_t = \tanh(U_h x_t + W_h (r_t \times h_{t-1}) + b_h) \tag{5}$$

where, U_h 、 W_h 、 b_h represents the weight parameters and offsets that candidate hidden layers need to learn.

Last, the hidden layer output of Bi-GRU can be expressed as

$$h_t = [\vec{h}_t, \overleftarrow{h}_t] \tag{6}$$

where, $\vec{h}_t$ represents the forward output of the hidden layer, $\overleftarrow{h}_t$ represents the directional output of the hidden layer, and h_t represents the output of the t-th GRU unit, $h_t \in R^{l \times d_{gru}}$ and $d_{gru} = 2 \times d$.

2.4 Context fusion

In order to better fuse the context information between sentence fragments and deeply explore the relationships between different event elements in complex events, this article uses the multi-heads self-attention mechanism to further fuse the context information. The structure is shown in Fig. 3.

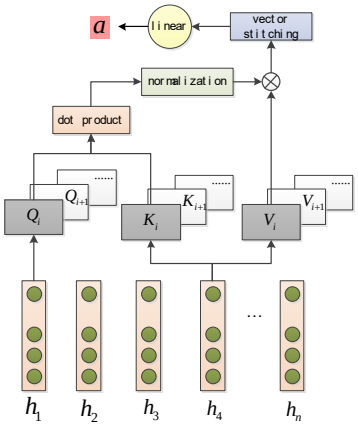

Fig. 3. Context fusion based on multi-heads self-attention mechanism

The multi-heads self-attention mechanism is an iterative evolution of the attention mechanism [23]. By performing multiple attention operations on the same text, and then splicing the results obtained by multiple attention heads, the attention tensor can be obtained, which can calculate the importance relationship between sequences from different feature dimensions. The calculation process is shown in equation:

$$a = concat(h_1, h_2, ..h_i, .., h_n)w \tag{7}$$

where, a represents the attention tensor, h_i represents the attention value obtained by the i-th attention head, and w is the weight matrix of the linearization process, $concat(\bullet)$ representing the direct splicing of vectors. The output calculation of the single head attention layer can be expressed as

$$h_i = Attention(QW_i^q, KW_i^k, VW_i^v) = soft\max(\frac{QW_i^q (KW_i^k)^T}{\sqrt{d_k}})VW_i^v \tag{8}$$

where, Q, K, V are the query vector matrix, key vector matrix, and value vector matrix, respectively. In this article, the initial values of the three are taken as the output values of the temporal coding layer, that is h, and W_i^q, W_i^k, W_i^v are the learnable parameter matrices of the i-th attention head, and d_k represents the dimension of the input vector. At the same time, Dividing $\sqrt{d_k}$ can reduce the impact of variance fluctuations on training.

2.5 Sequence callout

This article uses Conditional Random Field (CRF) to calculate tag probability. CRF can use a transition matrix to represent the correlation between tags, which can improve the classification effect of multiple tags compared to traditional softmax methods. The CRF loss function [24] is used to calculate model losses, and the model is optimized through back propagation. The specific formula is as follows:

$$loss = -\log \frac{P_{RealPath}}{P_1 + P_2 + ... + P_N} \tag{9}$$

where, $P_{RealPath}$ represents the true path score and P_N represents the N -th path score.

In order to solve the problem of dispersion and sharing of complex event elements, this paper proposes a method for defining complex event generic tags, which expands event arguments from the subject, object, trigger word, time, and location of ordinary event sentences to three major categories, that is, shared elements, domain elements, and others. The shared elements mainly include common subject, common object, common trigger, common location, and common time, while domain elements mainly include individual event elements, and others mainly include non-entity or unrelated characters. At the same time, event elements are labeled through the "B-I-O" method. The specific definition method is shown in Table 1, and the specific meaning and annotation examples of labels are respectively shown in Fig. 4 and Fig. 5.

Table 1

Complex event generic tags definition

	Subject label	Object label	Trigger label	Time label	Location label
share	B_sub/I_sub	B_obj/I_obj	B_tri/I_tri	B_tim/I_tim	B_loc/I_loc
domain	B_sub^1/I_sub^1	B_obj^1/I_obj^1	B_tri^1/I_tri^1	B_tim^1/I_tim^1	B_loc^1/I_loc^1
	B_sub^2/I_sub^2	B_obj^2/I_obj^2	B_tri^2/I_tri^2	B_tim^2/I_tim^2	B_loc^2/I_loc^2
	B_sub^3/I_sub^3	B_obj^3/I_obj^3	B_tri^3/I_tri^3	B_tim^3/I_tim^3	B_loc^3/I_loc^3
	B_sub^i/I_sub^i	B_obj^i/I_obj^i	B_tri^i/I_tri^i	B_tim^i/I_tim^i	B_loc^i/I_loc^i
other	O				

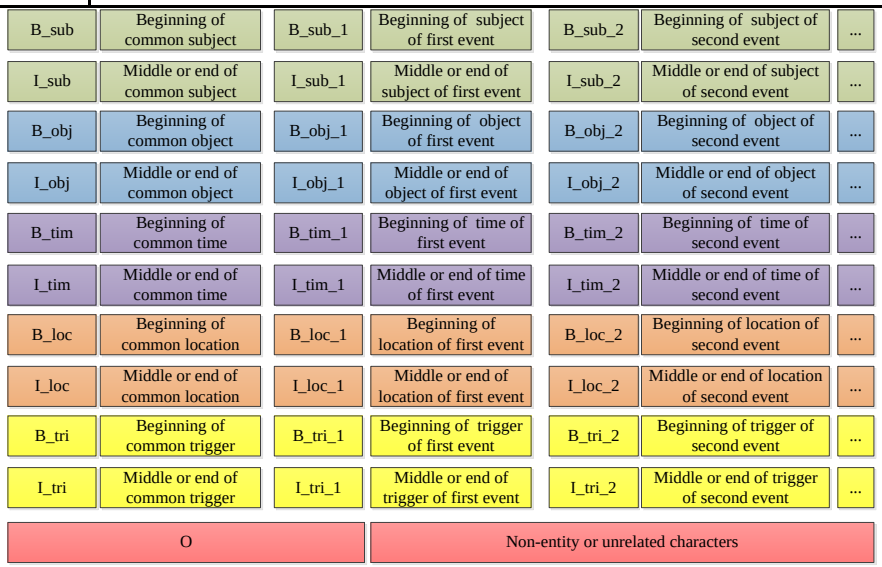

Fig. 4. Specific meanings of complex event generic tags

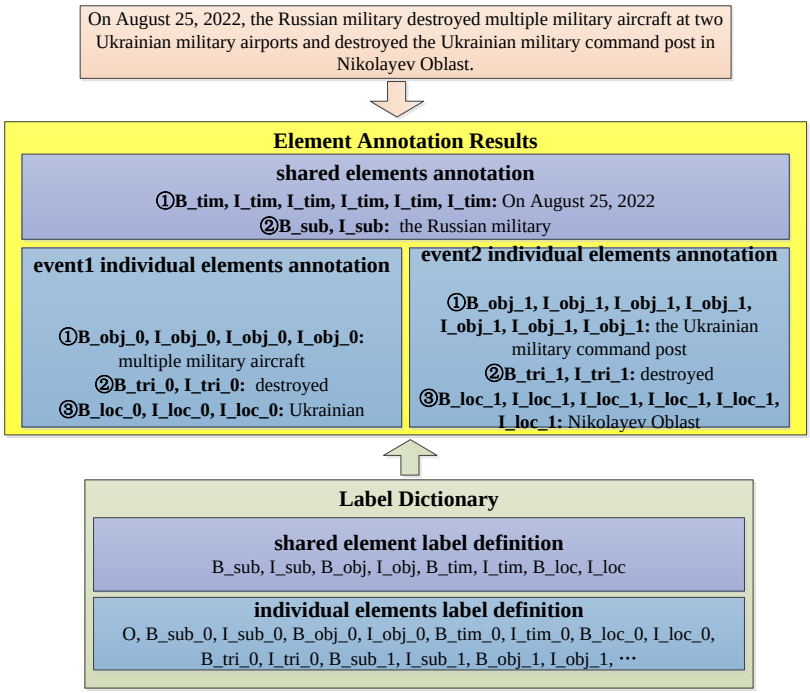

Fig. 5. Annotation instance of labels

By subdividing and defining complex event elements, using the "B-I-O" sequence annotation method and the context fusion complex event element extraction model proposed in this article, this article can effectively achieve the classification of shared elements and independent domain elements of complex event, decouple shared elements of different events, and simultaneously aggregate dispersed elements of the same events.

2.6 Algorithmic representation

To further clarify the steps and the implementation details, the paper provides algorithmic representation through pseudocode for easier replication in different fields, as shown in Fig. 6.

3. Results

The experimental data are sourced from mainstream news websites such as www.people.com,www.huanqiu.com and www.guancha.cn,etc. The content mainly includes economy, politics, diplomacy, military, security, science, etc. In order to verify the generalization ability of the model, there is no restriction on specific content of news, while fully considering the multiple complexities of complex events and insufficient samples.The experimental hardware environment consists of one NF5468M6 Inspur server with 256GB of memory, which includes two Intel Xeon Gold Medal 5318Y CPUs (with a main frequency of 2.1GHz) and

four Nvidia Graphics_ RTX3090 (24GB graphics memory). The software environment mainly includes Centos7.6, Python3.7, Pytorch1.10, and Cuda11.3.

input: sentence fragments[”sentence_1”,”sentence_2”,”sentence_3”⋯,”sentence_n”]

output: events elements[shared elements[common subject, common object, common trigger, common location, common time], domain elements[individual event elements],others[non-entity, unrelated characters]]

initialization: //initialize model hyperparameters

```
tf_block ← 12                 //number of transformer blocks
hidden_units ← 768            //hidden units of each transformer block
encoding_dim ← 768            //encoding dimensions of sentence fragments
opt_method ← Adam             //optimization method of model
learning_rate ← 1e-6          //learning rate of pre-trained model
fragments_max ← 5             //maximum number of sentence fragments
gru_layer ← 128               //number of hidden layer neurons in each direction
gru_init_rate ← 0.01          //initial learning rate of gru
gru_mid_rate ← 0.01 × (10/(number of iterations))   //learning rate of gru after 100 iterations
gru_dropout ← 0.8             //dropout probability of gru
attention_layer ← 1           //number of layer of attention
chain_len ← 512               //chain length of crf
train_length_max ← 128        //maximum length of the training sample
val_test_length_max ← 512     //maximum length of the validation and testing sample
gpu_sample ← 24               //number of used samples of each batches on each GPU
epochs ← 100                  //number of training sample traversals
training ← true               //training flag variable
validation ← false            //validation flag variable
test ← false                  //test flag variable
training_samples ← default    //80% of experimental samples
validation_samples ← default  //10% of experimental samples
testing_samples ← default     //10% of experimental samples
total_loss ← 0                //model loss, initial loss value is 0
while training do
    for i ← 1 to epochs do
        batches ← training_samples / (gpu_sample*4)   //four Nvidia Graphics_ RTX3090
        for j ← 1 to batches do
            e_j ← Roberta(tf_block, hidden_units, train_length_max, encoding_dim, learning_rate, gpu_sample)
            e ← [e_1, e_2, ..., e_j, ..., e_n]   //n = fragments_max
            h ← GRU(e, gru_layer, gru_init_rate, gru_mid_rate, gru_dropout, gpu_sample)   //capture temporal features
            h ← Attention(h, attention_layer)   //capture relevant features
            loss ← CRF(h, chain_len)   //calculate model losses
            backward(loss)   //back propagation
            optimizer(opt_method)   //update network parameters
            total_loss ← total_loss + loss   //accumulated loss value
        end
        save(model)   //save the trained model
    end
end
while validation do
    result ← model(validation_samples,val_test_length_max)
end
while test do
    result ← model(testing_samples,val_test_length_max)
end
```

Fig. 6. Algorithmic representation

This paper uses the commonly used accuracy, recall, and f1 values in the field of event element extraction to evaluate the algorithm model. The specific calculation method is as follows:

$$\begin{cases} Accurary = \dfrac{TP + TF}{TP + FP + FN + TN} \\ \mathrm{Re}call = \dfrac{TP}{TP + FN} \\ F1 = \dfrac{2 \times P \times R}{P + R} \end{cases} \tag{10}$$

where, TP represents that that the positive sample is predicted as a positive example, TF represents that the negative sample is predicted as a negative example, FP represents that the negative sample is predicted as a positive example, FN represents that the positive sample is predicted as a negative example.

The experimental validation data is manually filtered from the crawled news data to obtain complex event sentences without limiting the content. After being judged by three people and confirmed by two or more people to be reasonable, the data is retained. At the same time, the data is annotated according to the complex event element definition method proposed in this article, as shown in Fig. 5. The final experimental data is divided into two situations. One is a dataset containing 2784 complex event sentences, The second is a dataset containing 845 complex event sentences, which compares and verifies the effectiveness of the algorithm at different complexity and sample sizes in terms of complexity and quantity.

This paper first uses the dataset that contains 2784 complex event sentences, with a ratio of 8:1:1 for training, validation, and testing. The model parameters are set as follows. In character feature coding layer, transformer blocks are 12 and hidden units are 768. Encoding dimensions are 768 and optimization method is Adam. The learning rate of character feature encoding layer is 1e-6 and the maximum number of sentence fragments is set to 5. In timing sequence encoding layer, the number of GRU hidden layer neurons in each direction is 128 and the initial learning rate is 0.01. After 100 iterations, the learning rate is 0.01 × (10/(number of iterations)), and keeping the learning rate unchanged after 10000 iterations. The probability of dropout is 0.8. In context fusion layer, using a single layer of attention. In sequence callout layer, the chain length of CRF of is 512. Besides, the maximum length of the training sample is 128 and the maximum length of the validation and testing sample is 512, and the epochs are 100. The number of samples for both training and validation batches on each GPU is 24. The model's performance on event sentences of different complexity is verified, which mainly includes simple event sentences, one layer nested event sentences, and two layers nested event sentences. The main results are shown in Fig. 6- Fig. 9.

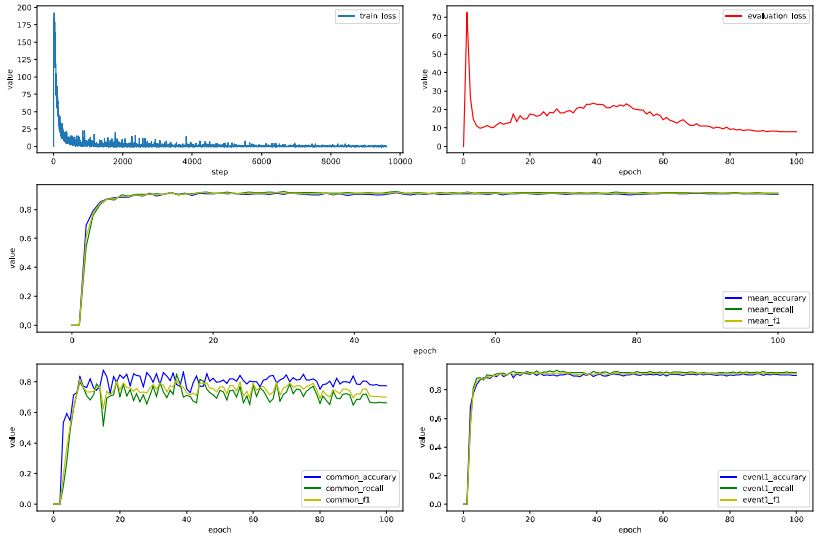

Fig. 6. Training and validation results of simple sentence element extraction

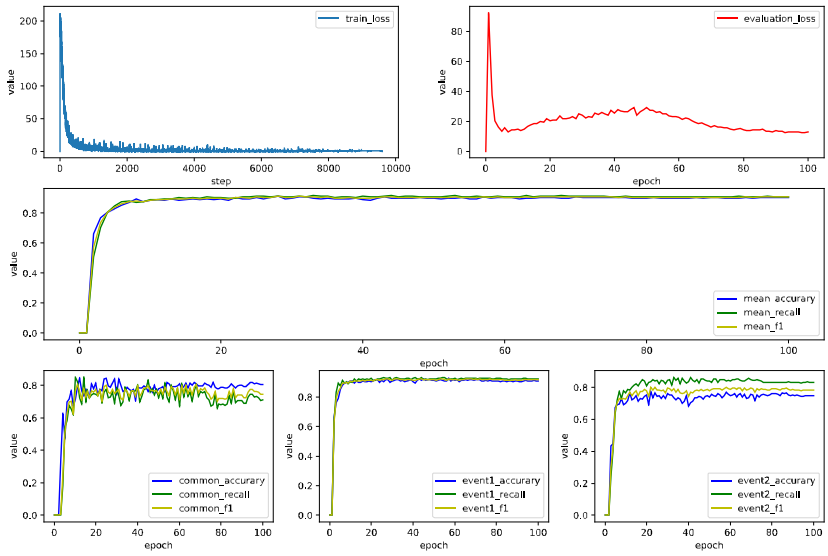

Fig. 7. Training and validation results of one layer nested event sentence element extraction

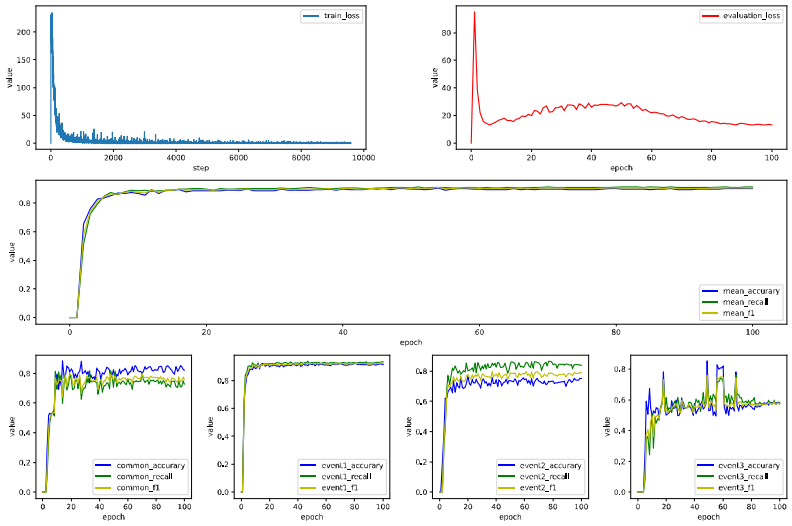

Fig. 8. Training and validation results of two layers nested event sentence element extraction

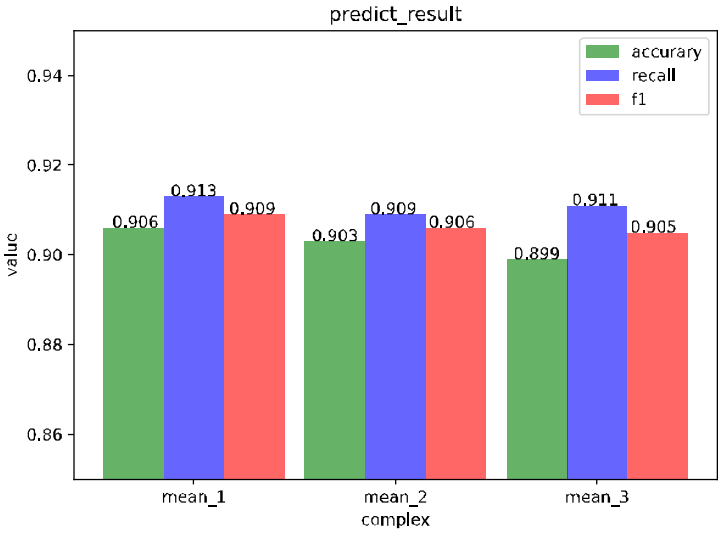

Fig. 9. Element extraction performance of complex event sentence

Through the experimental results in Figs. 6- Figs. 8, it can be found that the model tends to stabilize and maintain good performance in different complex situations such as simple sentences, one-layer nested sentence, and two layers nested sentence. At the same time, it can be found that the model achieves the best performance in approximately 60 epochs (5760 steps). Fig. 9 shows that the average accuracy, average recall rate and average f1 value of the model at different complexity are both around 0.9, indicating significant results.

At the same time, the model has strong adaptability to event complexity. The results of element extraction are shown in Table 2.

Table 2

Example of complex event extraction results

	Subject label	Object label	Trigger label	Time label	Location label	topic
text	On October 15th, during a meeting with the Chairman of the Foreign Affairs Committee of the French National Assembly, De Sarnez, Iranian Speaker Larijani relentlessly criticized European countries for not taking action to implement INSTEX and protested against French Total's withdrawal from Iran under pressure from US sanctions.					military
	The preliminary vote results released by the Spanish Ministry of the Interior in the early hours of the 24th showed that in the Spanish parliamentary elections held on the 23rd, the right-wing party People's Party became the largest party in parliament.					politics
	Positive progress was made in the new round of face-to-face negotiations between Russia and Ukraine held in Dolmabahçe Palace, Istanbul, Türkiye, on the 29th.					diplomacy
	In July 2023, Indian Prime Minister Modi announced that he would impose an export ban on various types of rice except for Indian fragrant rice.					security
	The three major indices of the US stock market rose together, with the Dow up 0.56%, the Nasdaq up 0.2%, and the S&P 500 up 0.44%.					economy
	On July 24th, according to foreign media reports, researchers in Arizona, USA recently launched a robot that can breathe, tremble, and sweat to study various changes.					science
share	Iranian Speaker Larijani	-	-	On October 15th	-	
	-	rose	-	-	-	
domain	-	European countries for not taking action to implement INSTEX	criticized	-	-	
	-	French Total's withdrawal from Iran under pressure from US sanctions	protested against	-	-	
	the Spanish Ministry of the Interior	the preliminary vote results	released	in the early hours of the 24th	-	
	the right-wing party People's Party	became	the largest party in parliament	on the 23rd	in the Spanish parliamentary elections	
	the new round of face-to-face	held	-	on the 29th	in Dolmabahçe Palace, Istanbul,	

	negotiations between Russia and Ukraine				Türkiye	
	Indian Prime Minister Modi	announced	he would impose an export ban on various types of rice except for Indian fragrant rice.	In July 2023	-	
	Dow	-	0.56%	-	-	
	Nasdaq	-	0.2%	-	-	
	the S&P 500	-	0.44%	-	-	
	researchers	launched	a robot that can breathe, tremble, and sweat to study various changes	On July 24th	in Arizona, USA	

As shown in Table 2, the event extraction capability of the model is evaluated in various topics as unstructured text, including military, politics, diplomacy, security, science. The results indicate that although the data description characteristics or content focus in different domains or topics are different, the paper proposes a universal label system, which reflects the differences in the specific annotation process and the model has relatively small differences in the extraction results of different domains or topics.

This paper uses the dataset that contains 845 complex event sentences, with a ratio of 8:1:1 for training, validation, and testing. The model parameters are consistent with the complexity verification experiment mentioned above, verifying the effectiveness of the model on complex event sentences with relatively small sample sizes. The extraction and prediction results of the two layers nested event sentence elements are shown in Fig. 10- Fig. 11.

Through the experimental results in Fig. 10, it can be found that the model can still achieve good results even when the sample size is small. Fig. 11 shows that in the case of a small sample size, the extraction effect decreases, and the average accuracy, average recall, and average f1 values of the model at different complexities reach above 0.8.

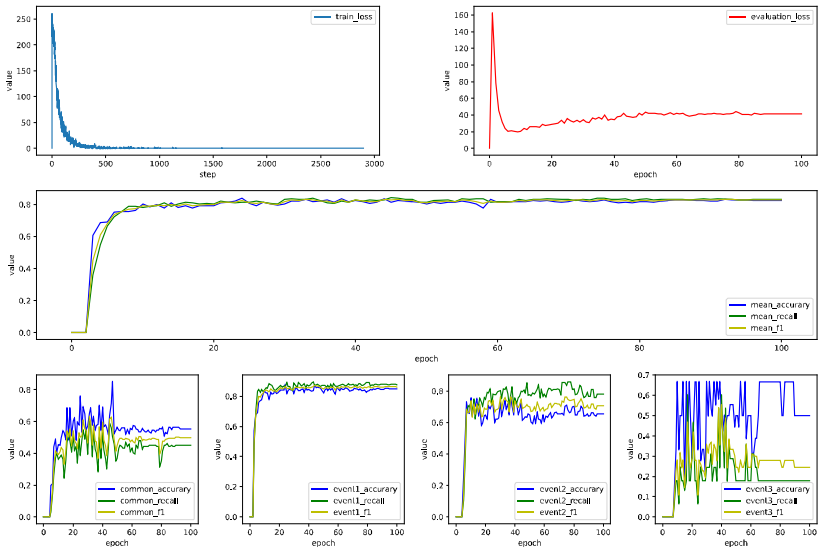

Fig. 10. Two layers nested event sentence element extraction training validation results (small sample size)

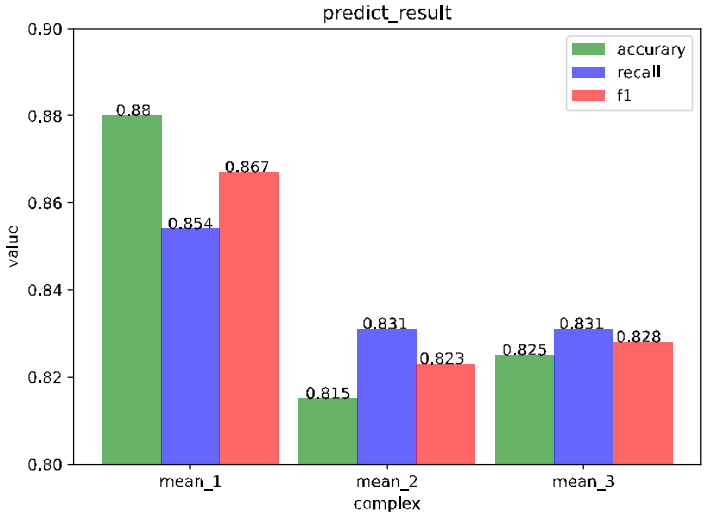

Fig. 11. Complex event sentence element extraction results (small sample size)

The values of the three are relatively close, and the effect is relatively stable, indicating that as the sample size increases, the model effect also enhances, and proving that the algorithm proposed in this paper has strong robustness to the complexity and sample size of complex events.

6. Conclusions

This article proposes a multi-domain oriented complex event element extraction algorithm with minimal domain and data content constraints. By

integrating pre-trained language models, temporal models, and attention mechanisms for encoding, complex event element decoupling and reorganization are achieved. Finally, experiments have verified the effectiveness and robustness of the algorithm under different complexity levels such as simple sentences, one-layer nested sentences, two layers nested sentences, and different sample sizes, which has a good reference and promoting effect on the extraction of complex event elements and document level element extraction.

However, the general labeling system of the proposed method is based on the characteristics of the event itself and is domain-independent, the impact of the labeling process is also significant, and the amount of data in different domains may vary by multiple orders of magnitude, especially for non-public domain data. The impact of data size on the model in this paper needs further research and exploration. Therefore, how to further unify annotation methods in different domains and explore the small sample capability of the model will be of great significance for research studies related to event extraction downstream tasks, such as event analysis, evaluation, prediction, etc.

REFERENCES

[1]. Qian Li, Jianxin Li, Jiawei Sheng, Shiyao Cui, Jia Wu, Yiming Hei, Hao Peng, Shu Guo, Lihong Wang, Amin Beheshti, et al., 2021a. A compact survey on event extraction: Approaches and applications. arXiv preprint arXiv:2107.02126.

[2]. Björne J, Ginter F, Pyysalo S, et al, Complex event extraction at PubMed scale[J]. Bioinformatics, 2010, 26(12): i382-i390.

[3]. KIM J T, MOLDOVAN D I, Acquisition of linguistic patterns for knowledge-based information extraction[J]. IEEE Transactions on Knowledge and Data Engineering, 1995, 7(5):713-24.

[5]. Kitani T, Eriguchi Y, Hara M, Pattern matching and discourse processing in information extraction from Japanese text[J]. Journal of Artificial Intelligence Research, 1994, 2: 89-110.

[6]. Wang W, Zhao D, Wang D, Chinese news event 5w1h elements extraction using semantic role labeling[C]//2010 Third International Symposium on Information Processing. IEEE, 2010: 484-489.

[7]. AHN D, The stages of event extraction[C]//Proceedings of the workshop on annotations and reasoning about Lime and events. Sydney, Australia: Association for Computational LinguisLics, 2006: 1-8.

[8]. LI Q, JI H, YU H, et a1, Constructing information networks using one single model[C]//Proceedings of the 2014 EMNLP. Doha, Qatar: ACL, 2014: 1846-1851.

[9]. WADDEN D, WENNBERU U, LUAN Y, et al, Entity, Relation, and Event Extraction with Contextualized Span Representations[C]//Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th international Joint Conference on Natural Language Processing(EMNLP-IJCNLP). Hong Kong, China: Association for Computational Linguistics, 2019:5784-5789.

[10]. Portelli B, Passabì D, Lenzi E, et al, Improving adverse drug event extraction with SpanBERT on different text typologies[M]//AI for Disease Surveillance and Pandemic

Intelligence: Intelligent Disease Detection in Action. Cham: Springer International Publishing, 2022: 87-99.
[11]. SI J H, PENG X T, LI C, et al, Generating Disentangled Arguments with Prompts: a Simple Event Extraction Framework that Works[EB/OL]. (2021-10-9)[2022-2-15]. http: //arxiv.org/abs/2110.04525.
[12]. Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 6759–6774, Dublin, Ireland. Association for Computational Linguistics.
[13]. Ting Jiang, Deqing Wang, Leilei Sun, Huayi Yang, Zhengyang Zhao, and Fuzhen Zhuang. 2021. Lightxml: Transformer with dynamic negative sampling for high-performance extreme multi-label text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 7987–7994.
[14]. Xinya Du, Sha Li, and Heng Ji. 2022. Dynamic global memory for document-level argument extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: 3171 Long Papers), pages 5264–5275, Dublin, Ireland. Association for Computational Linguistics.
[15]. Yaojie Lu, Qing Liu, Dai Dai, Xinyan Xiao, Hongyu Lin, Xianpei Han, Le Sun, and Hua Wu. 2022. Unified structure generation for universal information extraction. arXiv preprint arXiv:2203.12277.
[16]. Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022a. Dynamic prefix-tuning for generative template-based event extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Lin guistics (Volume 1: Long Papers), pages 5216–5228, Dublin, Ireland. Association for Computational Lin guistics.
[17]. Wang P, Deng Z, Cui R, Tdjee: A document-level joint model for financial event extraction[J]. Electronics, 2021, 10(7): 824.
[18]. Tong M, Xu B, Wang S, et al, DocEE: A large-scale and fine-grained benchmark for document-level event extraction[C]. Association for Computational Linguistics, 2022.
[19]. Shun Zheng, Wei Cao, Wei Xu, et al, Doc2EDAG: An end-to-end document-level framework for chinese financial event extraction. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing. 2019: 337-346.
[20]. Wang Z, Wang X, Han X, et al., CLEVE: Contrastive Pre-training for Event Extraction[J]. 2021.DOI:10.18653/v1/2021.acl-long.491.
[21]. Sha Li, Heng Ji and Jiawei Han. 2021b. Document-level event argument extraction by conditional generation[J]. arXiv preprint arXiv:2104.05919,2021.
[22]. Aim-Nang S, Seresangtakul P and Janyoi P. Isarn Dialect Word Segmentation using Bi-directional Gated Recurrent Unit with transfer learning approach 2022 26th International Computer Science and Engineering Conference (ICSEC). 10.1109/ICSEC56337.2022.10049346. 978-1-6654-9198-3. (156-160).
[23]. Xu X, Gao T, Wang Y, et al. Event Temporal Relation Extraction with Attention Mechanism and Graph Neural Network[J]. Journal of Tsinghua University: Natural Science Edition (English version), 2022, 27(1):12.
[24]. Sharma, Richa, S. Morwal, and B. Agarwal. Named entity recognition using neural language model and CRF for Hindi language[J]. Computer Speech & Language 74(2022):101356.