CLIO: Role-interactive Multi-event Head Attention Network for Document-level Event Extraction

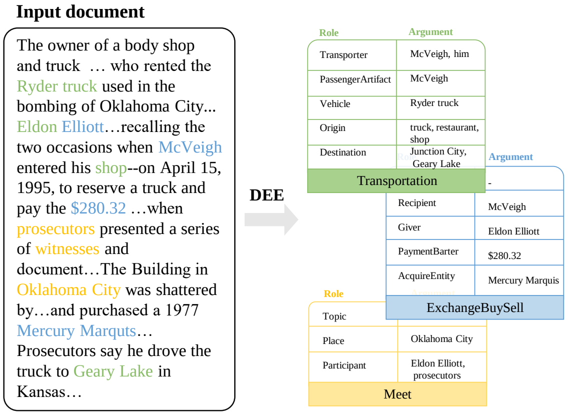
Yubing Ren[1,2], Yanan Cao[1,2], Fang Fang[1,2*], Ping Guo[1,2]
Zheng Lin[1,2], Wei Ma[1,2*] and Yi Liu[3]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]National Computer Network Emergency Response Technical Team/Coordination Center of China
{renyubing,caoyanan,fangfang0703,guoping,linzheng,mawei}@iie.ac.cn
liuyi@cert.org.cn

Abstract

Transforming the large amounts of unstructured text on the Internet into structured event knowledge is a critical, yet unsolved goal of NLP, especially when addressing document-level text. Existing methods struggle in Document-level Event Extraction (DEE) due to its two intrinsic challenges: (a) Nested arguments, which means one argument is the substring of another one. (b) Multiple events, which indicates we should identify multiple events and assemble the arguments for them. In this paper, we propose a role-interactive multi-event head attention network (CLIO) to solve these two challenges jointly. The key idea is to map different events to multiple subspaces (i.e., multi-event head). In each event subspace, we draw the semantic representation of each role closer to its corresponding arguments, then we determine whether the current event exists. To further optimize event representation, we propose an event representation enhancing strategy to regularize pre-trained embedding space to be more isotropic. Our experiments on two widely used DEE datasets show that CLIO achieves consistent improvements over previous methods.

1 Introduction

Cognitive scientists believe that humans remember and understand reality primarily in terms of events (Shipley and Zacks, 2008). Event studies are justifiably popular in Natural Language Processing (NLP), such as Event Coreference Resolution, Event Causality Identification, and Event Extraction. Event extraction is the process of extracting structured event knowledge from unstructured text and can be divided into sentence-level and document-level. Sentence-level Event Extraction has demonstrated promising results in empirical evaluations. However, in real-world scenarios, a large number of event elements are expressed across sentences. Document-level Event Extraction (DEE) is needed when we want to capture complete event information for the whole document. In contrast to SEE, increased text length brings more challenges, and DEE has still been underachieving.



The owner of a body shop and truck … who rented the Ryder truck used in the bombing of Oklahoma City… Eldon Elliott…recalling the two occasions when McVeigh entered his shop--on April 15, 1995, to reserve a truck and pay the $280.32 …when prosecutors presented a series of witnesses and document…The Building in Oklahoma City was shattered by…and purchased a 1977 Mercury Marquts… Prosecutors say he drove the truck to Geary Lake in Kansas…

Role	Argument
Transporter	McVeigh, him
PassengerArtifact	McVeigh
Vehicle	Ryder truck
Origin	truck, restaurant, shop
Destination	Junction City, Geary Lake
Transportation	

Role	Argument
	-
Recipient	McVeigh
Giver	Eldon Elliott
PaymentBarter	$280.32
AcquireEntity	Mercury Marquis
ExchangeBuySell	

Role	Argument
Topic	
Place	Oklahoma City
Participant	Eldon Elliott, prosecutors
Meet	

Figure 1: An illustration of DEE task. Different colored tabels indicate different event types. DEE needs to detect multiple event types and extract arguments for the roles of each event type.

Recently, researchers have shown an increased interest in DEE. Their works can be roughly divided into classification-based models (Zhang et al., 2020; Xu et al., 2021; Huang and Jia, 2021; Huang and Peng, 2021), tagging-based models (Yang et al., 2018; Du and Cardie, 2020), and generation-based models (Li et al., 2021; Yang et al., 2021; Du et al., 2021). The state-of-the-art approach (Liu et al., 2021) frames DEE as a machine reading comprehension task, assisted by two data augmentation regimes. Although scholars have made such valuable attempts in DEE, current methods still struggle in DEE due to the following crucial challenges:

Nested arguments: In a document, there are many nested arguments (i.e., one argument is the substring of another one) that belong to different roles. Figure 1 gives an example. In the "Trans-

[*]Fang Fang and Wei Ma are the co-corresponding authors

Proceedings of the 29th International Conference on Computational Linguistics, pages 2504–2514
October 12–17, 2022.

portation" event, "truck" (plays *Origin* role) and "Ryder truck" (plays *Vehicle* role) are nested event arguments. According to our statistics, 14.23% and 13.94% of documents in the WikiEvents (Li et al., 2021) and RAMS (Ebner et al., 2020) datasets have nested arguments, respectively. Unfortunately, these nested arguments can't be entirely identified by traditional tagging-based methods, which can not assign multiple labels to a token.

Multiple events: As shown in Figure 1, there are three kinds of events: "Transportation", "ExchangeBuySell", and "Meet" in a single document, and DEE should not only identify all events but also assign arguments to the corresponding events. The issue of multiple events is common in DEE (86.88% of documents in the WikiEvents involve multiple events). What's more, the arguments of these events are uniformly scattered across sentences, making it hard to achieve accurate arguments assembling. Previous works usually adopt a fixed document representation to detect all event types. However, different event types have different roles and arguments, and the emphasis of document representation should also be different.

For the nested arguments, which usually belong to different roles, the intuition is that we should extract arguments for each role independently. Assuming there are N roles in an event, we can perform N independent extractions by tagging arguments under each role. In this way, the argument substring "truck" of role "Origin" and the argument "Ryder truck" of role "Vehicle" can be identified at the same time. To address the challenge of multiple events, an intuitive way is to independently detect each event type and assemble arguments for it. For one event type, argument extraction can be simpler due to the decrease in roles. On the contrary, using role information specific to this event type can better detect the current event type. We argue that these two challenges can be solved jointly by mapping each event type to a specific subspace.

Analogy to multi-head attention (Vaswani et al., 2017), we propose a role-intera**C**tive mu**L**ti-event head attent**I**on netw**O**rk (CLIO) for DEE. The most critical part in CLIO is *Role-interactive Multi-event Head Attention* module, which can solve the aforementioned two challenges jointly. First, our attention module works in a role-centric way. That is to say, for each role, we extract all of its corresponding arguments independently. In this way, a token can be assigned multiple role labels, which can perfectly solve nested arguments problem. Second, our attention module assigns each event type a subspace by mapping it to each event head. In this way, we can independently detect each event type and assemble arguments for it, which can address the challenge of multiple events. In each event head, we use role information specific to this event to represent document. Such event-specific document representation eases the difficulty of detecting multiple events from a single document.

In summary, our contributions are as follows:

- We propose a role-interactive multi-event head attention network to handle the challenges of nested arguments and multiple events simultaneously.

- We conduct experiments on two widely used DEE datasets. Experimental results demonstrate that CLIO outperforms previous methods and has significant improvement when facing the vital challenges of DEE.

2 Methodology

We first describe the task formalization of DEE. Formally, given an input document comprised of m words $\mathcal{D} = \{w_i\}_{i=1}^{m}$, pre-defined event types $\mathcal{T} = \{t_i\}_{i=1}^{l}$, and role categorizies $\mathcal{R} = \{r_i\}_{i=1}^{n}$. The DEE task aims to extract one or more event records: $\{\text{event type} : t, r_1 : [a_1^1, a_1^2, ...], ..., r_i : [a_i^1, a_i^2, ...]\}$, where a_i^1 is the first argument of role r_i, and so on.

Figure 2 illustrates the architecture of CLIO, which consists of three key components: (1) Role-interactive Multi-event Head Attention, (2) Multiple Events Extraction, and (3) Event Representation Enhancing. *Role-interactive Multi-event Head Attention* module can solve the challenges of nested arguments and multiple events simultaneously.

2.1 Encoding

First, we construct an extended sequence $S = [\text{CLS}]\mathcal{D}[\text{SEP}]\mathcal{R}[\text{SEP}]$ by concatenating the document $\mathcal{D}$ and role set $\mathcal{R}$. Next, we use BERT (Devlin et al., 2019) with hidden size d to encode contextual embeddings of each word in the sequence S:

$$[\text{H}_w, \text{H}_r] = \text{BERT}(S) \quad (1)$$

After this stage, we can obtain the word representation of document $\text{H}_w \in \mathbb{R}^{m \times d}$ and role representation $\text{H}_r \in \mathbb{R}^{n \times d}$. This stage makes a deep fusion

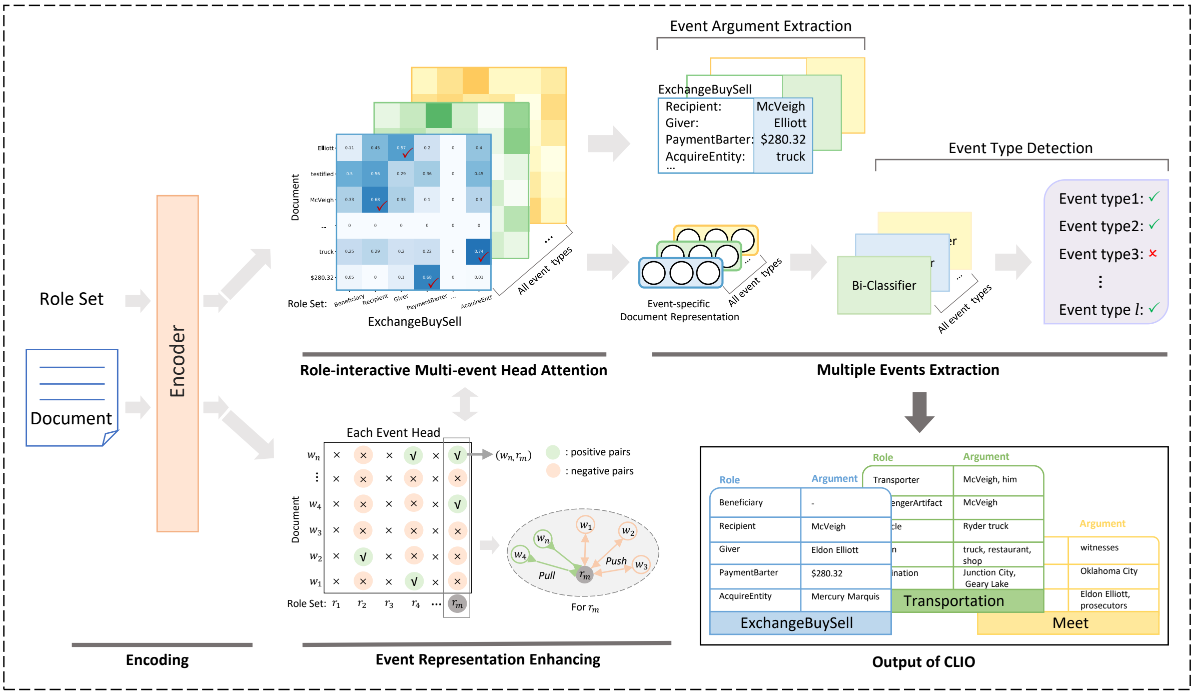


Figure 2: The overall architecture of CLIO. Role-interactive Multi-event Head Attention is designed to map each event type to a specific subspace. In each subspace, we compare the roles and words to measure the degree of relevance among them. Event Representation Enhancing is used to further optimize event representation. In Multiple Events Extraction, we perform the two subtasks of DEE.

between the document and roles by multi-head and multi-layer attention.

Norm-based Significance Score Intuitively, not every word in document is significant. So we introduce a norm-based significance score to measure the ability of words to express essential meaning based on the L2-Norm of word embedding. This feature of L2-Norm has already been proven by some promising works (Luhn, 1958; Chen et al., 2020; Liu et al., 2020).

We use the L2-Norm of word embeddings as the weight of them:

$$\mathrm{H}'_w = \|\mathrm{H}_w\|_2 \odot \mathrm{H}_w \tag{2}$$

where $\mathrm{H}'_w \in \mathbb{R}^{m\times d}$ is the weighted word embedding, $\odot$ means element-wise multiplication.

2.2 Role-interactive Multi-event Head Attention

In this step, the goal is to solve the challenges of nested arguments and multiple events simultaneously. We compare the role embeddings and word embeddings under each event type and select role-word pairs that have high semantic overlap as argument extraction results. We first consider a single event type, then extend it to all event types.

Role-interactive Event Attention

In each event type, we measure the degree of relevance between each role-word pair. We first project the original d-dimensional features of words and roles into a smaller dimension d' through two fully connected layers:

$$\begin{aligned} \tilde{\mathrm{H}}_w &= \mathrm{H}'_w \mathrm{W}_w + \mathrm{b}_w \\ \tilde{\mathrm{H}}_r &= \mathrm{H}_r \mathrm{W}_r + \mathrm{b}_r \end{aligned} \tag{3}$$

where $\mathrm{W}_w \in \mathbb{R}^{d\times d'}$, $\mathrm{b}_w \in \mathbb{R}^{d'}$, $\mathrm{W}_r \in \mathbb{R}^{d\times d'}$, $\mathrm{b}_r \in \mathbb{R}^{d'}$ are learnable parameters, $\tilde{\mathrm{H}}_w \in \mathbb{R}^{m\times d'}$, $\tilde{\mathrm{H}}_r \in \mathbb{R}^{n\times d'}$.

Then we apply concat attention (Luong et al., 2015) to measure the degree of relevance between word representation $\tilde{\mathrm{H}}_w$ and role representation $\tilde{\mathrm{H}}_r$. We indicate $\mathrm{S}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r)$ as the correlation intensity matrix of role-word pairs under the event type t:

$$\begin{aligned} \textsc{Score}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) &= \tanh([\tilde{\mathrm{H}}_w; \tilde{\mathrm{H}}_r]\mathrm{W}_a) \cdot v_a \\ \mathrm{S}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) &= \mathrm{sigmoid}(\textsc{Score}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r)) \end{aligned} \tag{4}$$

where $[\tilde{\mathrm{H}}_w; \tilde{\mathrm{H}}_r] \in \mathbb{R}^{m\times n\times 2d'}$, $\mathrm{W}_a \in \mathbb{R}^{2d'\times d'}$ and $v_a \in \mathbb{R}^{d'}$ are learnable parameters, $\textsc{Score}_t \in \mathbb{R}^{m\times n}$, $\mathrm{S}_t \in \mathbb{R}^{m\times n}$, $t \in \mathcal{T}$.

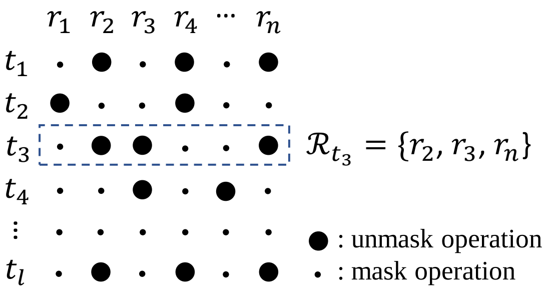

r_1 r_2 r_3 r_4 ⋯ r_n
t_1
t_2
t_3
t_4
⋮
t_l
$\mathcal{R}_{t_3} = \{r_2, r_3, r_n\}$
● : unmask operation
• : mask operation

Figure 3: Event schema mask **M**. t and r denote event type and role type, respectively. In each event type, we mask those roles not in the pre-defined role set.

Multi-event Head Attention

We perform the above role-interactive event attention on all event heads in parallel, which can extract multiple events simultaneously. Formally, we stack the role-interactive event attentions under all event types to a multi-event head attention $\mathrm{S}_{\mathcal{T}}(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) \in \mathbb{R}^{l \times m \times n}$, where the number of heads l is the size of event types.

For the DEE dataset, each event type t_i has a pre-defined role set $\mathcal{R}_{t_i}$ [1]. We formalize it as the event schema mask **M** (see Figure 3):

$$\mathbf{M}_{t_i, r_j} = \begin{cases} 1, & \text{role } r_j \text{ in } \mathcal{R}_{t_i} \\ 0, & \text{role } r_j \text{ not in } \mathcal{R}_{t_i} \end{cases} \tag{5}$$

Through the event schema mask **M**, we decrease the number of roles to predict under each event type, leave each event type a unique role candidate set and make a difference among event heads. The final multi-event correlation intensity matrix $\mathrm{S}^{multi}(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r)$ is caculated as:

$$\mathrm{S}^{multi}(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) = \mathrm{S}_{\mathcal{T}}(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) \odot \mathbf{M} \tag{6}$$

where $\mathrm{S}_{\mathcal{T}} \in \mathbb{R}^{l \times m \times n}, \mathbf{M} \in \mathbb{R}^{l \times n}, \mathrm{S}^{multi} \in \mathbb{R}^{l \times m \times n}$.

2.3 Multiple Events Extraction

Event Argument Extraction

The final multi-event correlation intensity matrix $\mathrm{S}^{multi}(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r)$ (Eq. 6) contains probabilities for each role-word pair. We take those role-word pairs whose probabilities are higher than threshold δ_{EAE} as the argument extraction results under the current role.

We use cross-entropy between the predictions and golden labels to optimize our model:

$$\mathcal{L}_{\text{EAE}} = \text{CE}(\mathrm{S}^{multi}, \mathrm{Y}^{multi}) \tag{7}$$

where $\mathrm{Y}^{multi} \in \mathbb{R}^{l \times m \times n}$ is the ground truth label for the correlation matrix between a document and roles under the multi-event head.

Event-specific Document Representation

To better detect which event type is contained in the document, we construct event-specific document representation for each event head.

Given the role-interactive event attention $\textsc{Score}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r)$ (Eq. 4) under event type t, we first normalize $\textsc{Score}_t$ with respect to role, referred to as A. We obtain word representation H''_w specific to event type t by using A to weighted sum the roles $\tilde{\mathrm{H}}_r$. Through the mean pooling operation we obtain document representation H_D specific to event type t:

$$\begin{aligned} A &= \text{softmax}_r(\textsc{Score}_t(\tilde{\mathrm{H}}_w, \tilde{\mathrm{H}}_r) \odot \mathbf{M}_{t,:}) \\ \mathrm{H}''_w &= A\tilde{\mathrm{H}}_r \\ \mathrm{H}_D &= \frac{1}{m} \sum_{w=w_1}^{w_m} \mathrm{H}''_w \end{aligned} \tag{8}$$

where $\textsc{Score}_t \in \mathbb{R}^{m \times n}$, $\mathbf{M}_{t,:} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\sum A_{i,:} = 1$, $\mathrm{H}''_w \in \mathbb{R}^{m \times d'}$, $\mathrm{H}_D \in \mathbb{R}^{d'}$.

Event Type Detection

We detect each event type based on the corresponding event-specific document representation. Concretely, we perform binary classification on H_D for event type t to get the probability P_t:

$$\mathrm{P}_t = \text{softmax}(\mathrm{H}_D \mathrm{W}_e) \tag{9}$$

where $\mathbf{W}_e \in \mathbb{R}^{d' \times 2}$ is learnable parameters, $\mathbf{P}_t \in \mathbb{R}^2$.

Then we expand P_t to the prediction of multiple events $\mathrm{P}^{event} \in \mathbb{R}^{l \times 2}$, which can identify multiple events simultaneously. We apply cross-entropy loss to update the model paremeters:

$$\mathcal{L}_{\text{ED}} = \text{CE}(\mathrm{P}^{event}, \mathrm{Y}^{event}) \tag{10}$$

where $\mathrm{Y}^{event} \in \mathbb{R}^{l \times 2}$ is the ground truth label for the event type.

2.4 Event Representation Enhancing

We find the language modeling of the above stages produces anisotropic word embeddings. So we apply intra-event contrastive learning to enhance event representation by regularizing pre-trained embedding space to be more isotropic. In DEE, we need to pull each role closer to its arguments (positives) while pushing each role away from other

[1] The pre-defined role sets are provided by DEE dataset.

words (negatives). Given a role, there are multiple arguments, i.e., there is more than one positive.

We apply an approach, proposed by (Hoffmann et al., 2022) based on InfoNCE, to include multiple positives. More specifically, for a role $\mathbf{h}_r \in \tilde{\mathrm{H}}_r$, words that are the arguments of role r form the set of positives $\mathcal{P}$, and words that are not the arguments of role r form the set of negatives $\mathcal{N}$, $\mathcal{P} \cup \mathcal{N} = \tilde{\mathrm{H}}_w$. To measure the similarity between a pair of features, we use the cosine similarity:

$$\mathrm{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}^\top \mathbf{y}}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|} \tag{11}$$

The training objective becomes:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\sum\limits_{\mathbf{p} \in \mathcal{P}} \exp(\frac{\mathrm{sim}(\mathbf{h}_r, \mathbf{p})}{\tau})}{\sum\limits_{\mathbf{p} \in \mathcal{P}} \exp(\frac{\mathrm{sim}(\mathbf{h}_r, \mathbf{p})}{\tau}) + \sum\limits_{\mathbf{n} \in \mathcal{N}} \exp(\frac{\mathrm{sim}(\mathbf{h}_r, \mathbf{n})}{\tau})} \tag{12}$$

where $\mathbf{h}_r \in \mathbb{R}^{d'}$ is the embedding of role r, $\mathbf{p} \in \mathbb{R}^{d'}$ is the argument embedding, and $\mathbf{n} \in \mathbb{R}^{d'}$ is the word embedding of the input document other than arguments , τ is a temperature hyperparameter.

2.5 Joint Learning

The overall loss function is divided into three parts: an event argument extraction loss $\mathcal{L}_{\text{EAE}}$, an event type detection loss $\mathcal{L}_{\text{ED}}$, and a contrastive loss $\mathcal{L}_{\text{CL}}$. We let these three objectives learn jointly at the same speed and update model parameters together. We have the following training loss:

$$\mathcal{L} = \lambda_1 \mathcal{L}_{\text{EAE}} + \lambda_2 \mathcal{L}_{\text{ED}} + \lambda_3 \mathcal{L}_{\text{CL}} \tag{13}$$

and λ_1, λ_2 are the weight dynamically adjusted with the training steps, where $\lambda_1 = \frac{1}{\mathcal{L}_{\text{EAE}}}, \lambda_2 = \frac{1}{\mathcal{L}_{\text{ED}}}$[2], λ_3 is hyperparameter.

3 Experiments

We evaluate our model's performance on the two commonly used DEE benchmarks and compare to prior work. Then we conduct an ablation study on how modules of our CLIO affect its performance on DEE task. We also conduct case study to analyze qualitatively the advantages and disadvantages of our model.

3.1 Experimental Setup

Datasets. We conduct our experiments on two widely used document-level event extraction datasets: RAMS (Ebner et al., 2020) and WikiEvents (Li et al., 2021). RAMS provides 9,124 annotated examples from news based on 139 event types and 65 roles. WikiEvents provides 246 annotated documents from news based on 50 event types and 59 roles. According to our statistics, 13.94% of documents in the RAMS have nested arguments. 14.23% and 86.99% of documents in the WikiEvents involve nested arguments and multiple events, respectively.

Evaluation Metrics. Our results are reported as Precision (P), Recall (R) and F-measure (F-1) score. Our argument extraction results are based on the Exact Match criterion: the predicted argument span should match exactly the gold one. As an event type often includes multiple roles, we use micro-averaged role-level scores as the final DEE metric.

Baselines. For strictly consistent comparison, we involve the following strong baselines:

- BERT-CRF (Loshchilov and Hutter, 2018), which combines BERT with Condition Random Field (Lafferty et al., 2001), is the most popular method in tagging-based event extraction.

- SpanSel (Ebner et al., 2020), which is based on span ranking, enumerates each possible span in a document to identify the most likely event arguments.

- Head-Expand (Zhang et al., 2020), which achieves state-of-the-art performance on the RAMS. It first identifies the head of an argument and then expands its region.

- BART-Gen (Li et al., 2021), which bases on the unfilled template and a given context, frames the implicit EAE as conditional generation.

- DocMRC (Liu et al., 2021), which frames DEE as Machine Reading Comprehension task, assisted by two data augmentation regimes.

Experimental Settings. We adopt pretrained BERT (Devlin et al., 2019) (bert-base-cased for English dataset), which has 12 hidden layers, each

[2] λ_1 and λ_2 only take the value of $\mathcal{L}_{\text{EAE}}$ and $\mathcal{L}_{\text{ED}}$, which contain no gradient information.

Methods	RAMS				WikiEvents			
	ED-F1	EAE-P	EAE-R	EAE-F1	ED-F1	EAE-P	EAE-R	EAE-F1
BERT-CRF (Loshchilov and Hutter, 2018)†	-	36.7	41.1	38.8	-	54.4	23.8	33.1
SpanSel (Ebner et al., 2020)†	-	38.0	38.4	38.2	-	**56.2**	26.2	35.7
Head-Expand (Zhang et al., 2020)†	-	-	-	40.1	-	55.4	25.4	34.8
BART-Gen (Li et al., 2021)†	-	20.7	30.3	24.6	-	14.2	7.8	10.1
DocMRC (Liu et al., 2021)	-	41.2	45.2	43.1	-	58.5	**30.5**	**40.1**
CLIO	**44.5**	**47.6**	**45.5**	**46.5**	**52.4**	48.7	29.5	36.8
w/o $\mathcal{L}_{\text{CL}}$	43.4	46.9	44.5	45.7	52.5	51.8	25.6	34.3
w/o norm	43.3	47.1	45.3	46.2	50	50.2	24.0	32.5

Table 1: The ED (Event Type Detection) and EAE (Event Argument Extraction) results of all models on the RAMS and WikiEvents datasets. Results marked † are from (Liu et al., 2021). DocMRC uses the expanded training data, **5 times** and **30 times** larger than RAMS and WikiEvents, to train its model. w/o $\mathcal{L}_{\text{CL}}$ and w/o norm denote we remove contrastive loss and norm-based significance weight respectively.

layer has 768 hidden units, and 12 attention heads. During training, we adopt mini-batch mechanism to train our model with batch size of 16, and the maximum training epoch is set to 100. We regularize our network using dropout, the dropout ratio of linear is 0.3. The initial learning rate is 2e-5 for BERT parameters and 2e-3 for other parameters. We trained all models with the AdamW optimizer (Loshchilov and Hutter, 2018). The warming up proportion for learning rate is 10%. Besides, the threshold δ_{EAE} for RAMS and WikiEvents is 0.6 and 0.65, respectively. We set temperature hyperparameter τ as 0.07. The weight of contrastive loss for RAMS and WikiEvents is 0.5 and 0.05, respectively.

In addition, the implementation of baselines does not consider gold event types. The experiments on RAMS consider event trigger information. We apply dot attention to measure the degree of relevance between role and trigger, and then we use the probability as the weight of role embedding.

3.2 Main Results

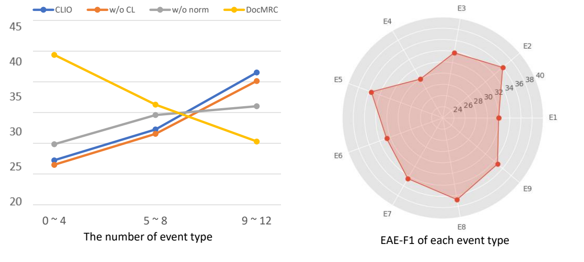
Table 1 presents our main results. Since the baselines do not have the capability for event type detection, the value of ED-F1 is replaced by '-'. We think event type detection is an integral part of DEE, while previous methods did not consider it. From Table 1, we can see that CLIO has the capability for event type detection. Our model surpasses all previous methods with 46.5 EAE-F1 score on the RAMS benchmark. Compared with the DocMRC, which uses 5 times more training data, our approach on the RAMS benchmark can bring substantial improvements in EAE, 3.4 F1 points. In the WikiEvents benchmark, our CLIO shows 3.0% drop in EAE-F1 scores compared to DocMRC. The reason is that DocMRC applies data augmentation, which expands training data to 30 times larger than WikiEvents. So it shows great advantages on small-scale dataset WikiEvents (expanding the original 246 documents to 7,491). Our CLIO reaches competitive results using only **1/30 data** compared with DocMRC. Compared with the SpanSel, our approach on the WikiEvents benchmark can bring 1.1 points of improvement in EAE-F1.

	RAMS		WikiEvents	
	Subset-N	Subset-O	Subset-N	Subset-O
DocMRC	40.6	43.4	39.9	**41.3**
CLIO	**47.1**$_{\uparrow 6.5}$	**44.0**$_{\uparrow 0.6}$	**46.2**$_{\uparrow 6.3}$	32.6

Table 2: Overall EAE-F1 with nested argument handling. Subset-N is a nested subset, while Subset-O is a non-nested subset.

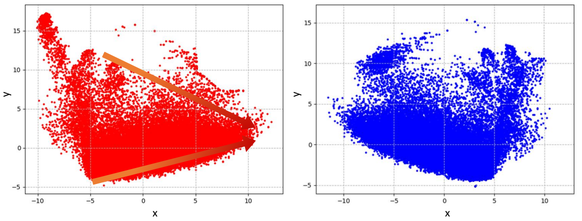
CLIO can extract nested arguments accurately. We conduct additional experiments to evaluate the capability of CLIO to extract nested arguments. The idea is to split the test data into two portions: documents with and without nested arguments (Subset-N and Subset-O). Table 2 shows the results. On the Subset-N, CLIO significantly outperforms DocMRC by 6.5 F1 and 6.3 F1 on the RAMS and WikiEvents, respectively. We conclude that CLIO achieves superior performance in both datasets largely because it solves the nested argument issue.

CLIO can handle complex multi-event scenarios. Figure 4 presents the additional experiment results. From (a), we can observe that as the number of event types increases, the performance of CLIO increases instead of decreases while the per-



(a) (b)

Figure 4: (a) EAE-F1 scores on the WikiEvents with different numbers of event types. (b) Radar chart of the average EAE-F1. “E1, E2,... E9” are the event types CLIO identified. The red dot is the micro EAE-F1 on the role level.



(a) CLIO w/o CL (b) CLIO

Figure 5: 2D visualization of projected word embeddings[3]. (a). Word embeddings trained by CLIO (without contrastive learning). (b). Word embeddings trained by CLIO.

formance of DocMRC decreases, which indicates CLIO can handle complex multi-event scenarios. In (b), we randomly choose a document from the test set of WikiEvents and calculate the EAE-F1 under each event type. We find the EAE-F1 values evenly distributed on all event types, which indicates CLIO has the capability to handle multiple events.

3.3 Ablation Study

We perform an ablation study to test how useful our event representation enhancing and norm-based significance weight. The results are shown in Table 1. Specifically, “w/o $\mathcal{L}_{\text{CL}}$” denotes contrastive loss is not considered in the joint learning, “w/o norm” means word embeddings are not weighted with norm significance score. In Figure 5, we compare the 2D visualization of word embeddings with or without contrastive learning. We can see that the event representation enhancing strategy can alleviate the representation degeneration problem and improve the isotropic properties of these representations.

3.4 Case Study

We present two examples from both datasets to illustrate the capability of CLIO. The examples are presented in Table 3, including input document and event extraction results. From Table 3, we find that CLIO can help DEE in two ways:

Handling nested arguments accurately In the first example, “U.S.” and “U.S. officials” are nested arguments, belonging to the role “place” and the role “communicator” respectively. CLIO works in a role-centric way, which can extract both of them together and assign them to corresponding roles. This case demonstrates how role-interactive event attention can assign multiple role labels to each token and solve the challenges of nested arguments.

Handling complex multi-event scenarios In the second example, “E1”, “E2”, and “E3” indicate three different event types. In multi-event scenario, CLIO can not only identify all events but also assign arguments to the corresponding events. CLIO assigns each event type a subspace by mapping it to each event head, where the event-specific document representation eases the difficulty of detecting multiple events. This implies that CLIO is particularly helpful for the extraction of multi-event scenarios.

4 Related Work

Sentence-level Event Extraction SEE extracts the event trigger and its arguments from a single sentence. Researchers have made a lot of progress in this field. Li et al. (2013, 2015) employ various hand-designed features to extract event; (Nguyen and Grishman, 2015; Nguyen et al., 2016; Chen et al., 2015; Liu et al., 2017, 2018) use neural based models such as recurrent neural networks (Zaremba et al., 2014) and convolutional neural network (LeCun et al., 1998) to extract event. With the recent success of BERT (Devlin et al., 2019), pretrained language models have also been used for SEE (Wang et al., 2019b,c; Yang et al., 2019; Wadden et al., 2019; Tong et al., 2020; Wang et al., 2021; Lu et al., 2021; Liu et al., 2022). These approaches achieve remarkable performance in benchmarks such as ACE 2005 (Walker et al., 2005) and similar datasets (Ellis et al., 2015; Ji et al., 2016; Getman et al., 2017).

Document-level Event Extraction Different from SEE, DEE does not need to explicitly recognize event triggers. The goal of DEE is to iden-

[3] Note that we project the original word embeddings to a 2-dimensional vector space using principal component analysis (PCA) for the purpose of visualization.

Category	Example
Nested arguments	**Ex1.** From the media we discovered that some local authorities we approached coordinated their negative decision with the federal government...Reporters at the State Department 's daily press briefing on Friday asked if [[**U.S.**]$_{\textbf{place}}$**officials**]$_{\textbf{communicator}}$ had advised [**individual states**]$_{\textbf{recipient}}$ not to allow in Russian observers. (**Event type**: contact.requestadvise.correspondence)
Multiple events	**Ex2.** Japanese [**police**]$_{\textbf{E1-Jailer}}$ have arrested a [**man**]$_{\textbf{E1-Detainee}}$ who admitted to landing a drone with low-level radioactive sand on the roof of the prime minister's office...Tokyo metropolitan police said [**Yasuo Yamamoto**]$_{\textbf{E1-Detainee}}$, 40, turned himself in to authorities late Friday in Fukui in western Japan...The small [**drone**]$_{\textbf{E2-IdentifiedObject}}$ found Wednesday had traces of radiation and triggered fears of potential terrorist attacks using [**unmanned aerial devices**]$_{\textbf{E3-Instrument}}$...

Table 3: Case study on the RAMS (Ex1) and WikiEvents (Ex2) test sets. The bold text indicates the argument word. Predicted arguments are marked with [square brackets] span indicator. Ex2 includes multiple events, where E1: Justice.ArrestJailDetain, E2: Cognitive.IdentifyCategorize, E3: Conflict.Attack.

tify event types and extract arguments of roles from the whole document. On the task level, most of these works fall into three categories: (1) classification-based models (2) tagging-based models (3) generation-based models. Zhang et al. (2020); Xu et al. (2021); Huang and Jia (2021); Huang and Peng (2021) employ traditional classification paradigm to determine the event type, then they identify the arguments and classify the roles they play in an event; Yang et al. (2018); Du and Cardie (2020) use the sequence labeling model BiLSTM (Zhang et al., 2015) -CRF (Lafferty et al., 2001) to automatically extract events; Li et al. (2021) frame the problem as conditional generation. Yang et al. (2021) apply cross attention mechanism to extract structured events in a parallel manner. Above methods conduct experiments on MUC-4 (McLean, 1992), WikiEvents (Li et al., 2021), RAMS (Ebner et al., 2020), and Chinese financial dataset (Zheng et al., 2019).

Contrastive Learning In NLP, contrastive self-supervised learning has been widely used for learning better representations by contrasting positive pairs and negative pairs. The core idea is to concentrate positive samples while pushing apart negative samples. InfoNCE (Oord et al., 2018) is a frequently used objective function in contrastive learning. It maximizes the similarity of positive pairs and minimizes the similarity of negative pairs. More specifically, for a query q, a single positive p and a set of negatives $\mathcal{N} = \{n_1, ..., n_k\}$ is given. To measure the similarity between a pair of features, it uses the cosine similarity as the training objective:

$$\mathcal{L}_{\text{InfoNCE}} = -\log \frac{\exp(\frac{\text{sim}(q,p)}{\tau})}{\exp(\frac{\text{sim}(q,p)}{\tau}) + \sum_{n \in \mathcal{N}} \exp(\frac{\text{sim}(q,n)}{\tau})} \tag{14}$$

Anisotropy Gao et al. (2019); Wang et al. (2019a) have pointed out that language modeling usually produces an anisotropic word embedding space. This phenomenon is also observed in the pretrained Transformers like BERT, GPT-2, etc (Ethayarajh, 2019). Li et al. (2020) thinks that "anisotropic" means word embeddings occupy anarrow cone in the vector space. Through empirical analysis, we find that the word representations in documents have high cosine similarity between each other, which is known as anisotropic word embeddings. In a document, some words are event arguments while others are event-irrelevant, which means they should not learn similar word representations.

5 Conclusion

In this paper, we propose a role-interactive multi-event head attention network (CLIO) for DEE. By mapping different events to multiple subspaces, we decomposed DEE into multiple substeps to handle nested arguments and multiple events. To further optimize event representation, we apply an event representation enhancing strategy to regularize pre-trained embedding space to be more isotropic. Experimental results show that CLIO can significantly outperform previous methods, especially when facing the specific challenges of DEE. In future work, we would like to explore superior word representation specific to events.

Acknowledgements

This work is supported by the Youth Innovation Promotion Association of the Chinese Academy of Sciences (No.2018192), and the NSFC (No.61902394). We would like to thank the anonymous reviewers for their valuable comments.

References

Kehai Chen, Rui Wang, Masao Utiyama, and Eiichiro Sumita. 2020. Content word aware neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 358–364, Online. Association for Computational Linguistics.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176, Beijing, China. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020, Online. Association for Computational Linguistics.

Xinya Du, Alexander Rush, and Claire Cardie. 2021. Template filling with generative transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 909–914, Online. Association for Computational Linguistics.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8057–8077, Online. Association for Computational Linguistics.

Joe Ellis, Jeremy Getman, Dana Fore, Neil Kuster, Zhiyi Song, Ann Bies, and Stephanie M Strassel. 2015. Overview of linguistic resources for the tac kbp 2015 evaluations: Methodologies and results. In *TAC*.

Kawin Ethayarajh. 2019. How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 55–65, Hong Kong, China. Association for Computational Linguistics.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2019. Representation degeneration problem in training natural language generation models. *arXiv preprint arXiv:1907.12009*.

Jeremy Getman, Joe Ellis, Zhiyi Song, Jennifer Tracey, and Stephanie M Strassel. 2017. Overview of linguistic resources for the tac kbp 2017 evaluations: Methodologies and results. In *TAC*.

David T Hoffmann, Nadine Behrmann, Juergen Gall, Thomas Brox, and Mehdi Noroozi. 2022. Ranking info noise contrastive estimation: Boosting contrastive learning via ranked positives. *arXiv preprint arXiv:2201.11736*.

Kung-Hsiang Huang and Nanyun Peng. 2021. Document-level event extraction with efficient end-to-end learning of cross-event dependencies. In *Proceedings of the Third Workshop on Narrative Understanding*, pages 36–47, Virtual. Association for Computational Linguistics.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 340–351, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Heng Ji, Joel Nothman, H Trang Dang, and Sydney Informatics Hub. 2016. Overview of tac-kbp2016 trilingual edl and its impact on end-to-end cold-start kbp. *Proceedings of TAC*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. *arXiv preprint arXiv:2011.05864*.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 894–908, Online. Association for Computational Linguistics.

Xiang Li, Thien Huu Nguyen, Kai Cao, and Ralph Grishman. 2015. Improving event detection with abstract meaning representation. In *Proceedings of the first workshop on computing news storylines*, pages 11–15.

Jian Liu, Yufeng Chen, and Jinan Xu. 2021. Machine reading comprehension as data augmentation: A case study on implicit event argument extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2716–2725, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Xuebo Liu, Houtim Lai, Derek F. Wong, and Lidia S. Chao. 2020. Norm-based curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 427–436, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2018. Fixing weight decay regularization in adam.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Hans Peter Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Virginia McLean. 1992. Fourth message understanding conference (muc-4).

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309, San Diego, California. Association for Computational Linguistics.

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*.

Thomas F Shipley and Jeffrey M Zacks. 2008. *Understanding events: From perception to action*, volume 4. Oxford University Press.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Christopher Walker, Stephanie Strassel, Julie Medero, and Kazuaki Maeda. 2005. Ace 2005 multilingual training corpus-linguistic data consortium. *URL: https://catalog. ldc. upenn. edu/LDC2006T06*.

Lingxiao Wang, Jing Huang, Kevin Huang, Ziniu Hu, Guangtao Wang, and Quanquan Gu. 2019a. Improving neural language generation with spectrum control. In *International Conference on Learning Representations*.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019b. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008, Minneapolis, Minnesota. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019c. HMEAE: Hierarchical modular event argument extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5777–5783, Hong Kong, China. Association for Computational Linguistics.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: Contrastive Pre-training for Event Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online. Association for Computational Linguistics.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3533–3546, Online. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6298–6308, Online. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

Wojciech Zaremba, Ilya Sutskever, and Oriol Vinyals. 2014. Recurrent neural network regularization.

Shu Zhang, Dequan Zheng, Xinchen Hu, and Ming Yang. 2015. Bidirectional long short-term memory networks for relation classification. In *PACLIC*.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7479–7485, Online. Association for Computational Linguistics.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. 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 (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.