Explicit Role Interaction Network for Event Argument Extraction

Nan Ding[1], **Chunming Hu**[1,2*], **Kai Sun**[1], **Samuel Mensah**[3], **Richong Zhang**[1,2]
[1]SKLSDE, Beihang University, Beijing, China
[2]Zhongguancun Laboratory, Beijing, China
[3]Department of Computer Science, University of Sheffield, UK
{dingn, hucm, sunkai}@buaa.edu.cn
s.mensah@sheffield.ac.uk, zhangrc@act.buaa.edu.cn

Abstract

Event argument extraction is a challenging subtask of event extraction, aiming to identify and assign roles to arguments under a certain event. Existing methods extract arguments of each role independently, ignoring the relationship between different roles. Such an approach hinders the model from learning explicit interactions between different roles to improve the performance of individual argument extraction. As a solution, we design a neural model that we refer to as the Explicit Role Interaction Network (ERIN) which allows for dynamically capturing the correlations between different argument roles within an event. Extensive experiments on the benchmark dataset ACE2005 demonstrate the superiority of our proposed model to existing approaches.[1]

1 Introduction

Event extraction is an important research field in information extraction, and its purpose is to extract structured information describing events from natural language texts. Event extraction includes two sub-tasks, namely (1) event detection (ED): trigger recognition and event type classification, and (2) event argument extraction (EAE): argument identification and role classification. For example, given a sentence "A group of soldiers were attacked on Friday.", it describes an *Attack* event, triggered by the word "attacked" and accompanied by 2 arguments: "A group of soldiers" and "Friday", which play the role of *Victim* and *Time* respectively. Event detection is a prerequisite for event extraction. The technologies and methods in this task are relatively established, and there has been great advancement in recent years (Li et al., 2021; Cui et al., 2020; Lai et al., 2020). However, the progress in event argument extraction is still

The US president[Person] first visited Japan since his appointment [Start Position] showing ties between Tokyo and Washington[Entity].		
w/o Interaction	Tokyo	×
w/ Interaction	Washington	√

(a) The *Person* argument "US president" helps exclude the interfering entity "Tokyo" and implies the golden argument "Washington" of *Entity*.

The building in the capital[Place][Target] took a direct hit[Attack], glass across the street in the neighborhood[Place].		
w/o Interaction	neighborhood	Partially√
w/ Interaction	neighborhood,capital	Completely√

(b) By informing model of the *Target* argument "The building in the captital", it is easier to identify the overlapping argument "capital".

Figure 1: Two examples of EAE demonstrating the importance of role interaction. The trigger words, followed by its event type in "[]", are marked in green. The arguments to be extracted are marked in red, followed by its role type in "[]" while the relevant arguments for interaction are marked in blue.

far from satisfactory with F1 score of about 55%, which is much lower than that of event detection (Li et al., 2020; Ma et al., 2020; Zhang et al., 2020). This identifies event argument extraction as a major bottleneck for event extraction. Hence in this paper we focus on the event argument extraction task.

Most existing methods for event argument extraction are neural network based, modeling the task as a sequence labeling problem and focusing on learning expressive features from text (Nguyen and Nguyen, 2019; Zhang et al., 2019b; Lin et al., 2020). An issue that has to be addressed for those methods is that arguments of different roles for an event may overlap with each other. Thus, it is not practical to extract all arguments simultaneously. As a consequence, majority of approaches are designed to extract arguments of each role independently (Ma et al., 2020; Zhang et al., 2020; Yang et al., 2019).[2] However, such an approach

*Corresponding author

[1]Our code is available at `https://github.com/bellytina/Explicit_Role_Interaction_Network`

[2]Note that two overlapping arguments will not play the

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 3475 - 3485
December 7-11, 2022 ©2022 Association for Computational Linguistics

addresses arguments of each role in isolation, hindering model from learning the correlation between roles.

We argue that it is important to consider role interactions for event argument extraction. On the one hand, the arguments of one role may closely relate to the arguments of another role. Entities that interfere with the argument to be extracted can be excluded by using information from other related arguments. As an example shown in Figure 1(a), when identifying the arguments of *Entity* for the *Start Position* event, "Tokyo" and "Washington" both have a chance of becoming candidate arguments. The introduction of interaction with the *Person* argument "US president" will provide clues for the model to identify the correct argument "Washington" since "Washington" is the semantically relevant argument with "US". Meanwhile, the model without the interaction wrongly extracts the "Tokyo" as the argument.

On the other hand, the arguments of some related roles may overlap with each other. Learning the interaction between those overlapping arguments can help improve the individual extraction of those arguments. As an example shown in Figure 1(b), the *Target* argument "The building in the capital" and the *Place* argument "capital" share a nested structure. Both *Target* and *Place* roles indicate the location-related argument. The model without interaction fails to identify the inner argument "capital". By introducing the interaction with the *Target* argument, the model will be guided to pay more attention to content in that text span of the *Target* argument, since there might be another location-related argument within the span. Then the model successfully extract the "captital" argument.

Based on these observations above, we propose a novel Explicit Role Interaction Network (ERIN) for event argument extraction. The main contribution in ERIN is the proposed Role Interaction Module (RIM) which serves to dynamically capture the correlations between different roles in an event. Specifically, RIM learn interactions in an explicit way by considering prediction outputs of argument extraction, allowing argument spans to be mutually aware and establishing an information transmission between roles. Besides, a Transformer in RIM enables the learning of a global interaction across roles. The RIM is also designed as a multi-layer architecture to match the requirement for gradually learning more complicated interaction cues, resulting in more refined results of argument extraction. To verify the effectiveness of our model, we conduct extensive experiments on the benchmark ACE2005 dataset. Our model attains state-of-the-art performance, and improves the F1 score of argument extraction to 57.96% (+2.96%).

same role, otherwise they will be merged into one argument.

2 Related Work

The mainstream methods for event extraction can be categorized into two classes: the pipeline models (Zhang et al., 2020; Liu et al., 2020; Yang et al., 2019; Li et al., 2020; Du and Cardie, 2020), which feeds results of ED to the downstream task EAE, and the joint models (Ma et al., 2020; Sha et al., 2018; Lu et al., 2021), which extracts triggers and arguments simultaneously. Under the pipeline approaches, (Zhang et al., 2020; Liu et al., 2020; Li et al., 2020; Du and Cardie, 2020) are question answering based models that search answers to questions as results of extraction. (Yang et al., 2019) conducts data augmentation for insufficient training corpus. Among joint models, (Sha et al., 2016) and (Sha et al., 2018) propose to leverage interactions between arguments. (Ma et al., 2020) incorporates both event-specific and syntactic information. (Lu et al., 2021) transforms text sequences into event structures. Some methods also integrate named entity recognition (NER) as additional supervision (Nguyen and Nguyen, 2019; Wadden et al., 2019; Zhang et al., 2019b; Lin et al., 2020; Zhang et al., 2019a). Though joint models alleviate the error propagation existing in pipeline-based methods, designing the optimal joint way of interdependence between two tasks is complex, which frequently results in a substandard architecture that is difficult to optimize (Lu et al., 2021). Our work follows the pipeline-based approaches, but focuses on modeling interactions between roles for event argument extraction.

So far, a majority of works have been conducted to improve performance in many fields by exploiting explicit interactions between different kinds of information in the task, such as relation extraction (Sun et al., 2020), sentiment analysis (He et al., 2019) and NER (Zhao et al., 2019), while only few works have explored interactions between roles for event argument extraction (Sha et al., 2018, 2016). Among these works, Sha et al. (2016) define the interaction as positive and negative correlation, which only determines whether or not this argument is to be recognized. (Sha et al., 2018) calculate the interaction vectors between each pair of candidate arguments, and add a self-matching matrix to judge whether two arguments tend to occur together. However, these works focus on learning interactions based on intermediate representations of networks, ignoring the interdependence between prediction outputs of argument extraction. In this paper, we consider the classifier outputs of argument extraction to provide an explicit supervision signal for model to learn the interactions.

3 Problem Definition

In this section, we formally describe the event argument extraction (EAE) task. Given an event text $X = \{x_1, x_2, \ldots, x_n\}$ of n words, the event trigger t is a text span in X, which indicates an event type $e_t \in T$, where $T = \{e_1, ..., e_u\}$ is the set of all u different event types. Given a triple (X, t, e_t), the goal of EAE is to extract all the arguments related to t in X, and assign a role $r \in R_t$ to each extracted argument a_t, where $R_t = \{r_1, ..., r_m\}$ is the set of m different roles associated with the event type e_t, and the argument a_t is a text span in X. The event schema is usually pre-defined, and each type of event contains a set of roles.

4 Model

In this section, we introduce our proposed Explicit Role Interaction Network (ERIN) for event argument extraction. The overview of the network is shown in Figure 2. First we introduce the text encoder of ERIN which we refer as Role-Aware Encoder (RAE). Next we introduce the proposed Role Interaction Module (RIM) which is designed to capture the correlations between different roles in an event. Then we introduce the training objective of ERIN and briefly present our model for event detection.

4.1 Role-Aware Encoder

The EAE task is formulated as a sequence labeling problem. Given an input sequence $X = \{x_1, x_2, \ldots, x_n\}$ of n words and an event trigger t with its type e_t, the goal is to predict the label sequence $Y = \{y_i\}_{i=1}^{n}$, with $y_i \in \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$, denoting the **B**eginning, **I**nside and **O**utside of an argument a_t that relates to trigger t. Since the text spans of arguments that relate to different roles may overlap with each other, it is not practical to extract all arguments simultaneously. Thus, we develop a Role-Aware Encoder (RAE) to incorporate the role information when encoding the text, aiming to specify the role of the arguments that are to be extracted from the text.

Following previous works (Ma et al., 2020; Liu et al., 2020), we adopt the pre-trained BERT model (Devlin et al., 2019) as the text encoder. To integrate the role information, we pair the text description of a role $r_i \in R_t$ with the input text X to form a new input sequence "[CLS] X [SEP] r_i [SEP]". Besides, to make full use of the information of the given event, we additionally enhance the BERT embeddings with two types of initial embeddings: (1) event type embeddings evt, which emphasizes the occurrence of the event scene; (2) position embeddings pst which identify the relative distance from each word $x_i \in X$ to the trigger words t. We lookup both embeddings of event types and positions in a randomly initialized embedding table. Then evt, pst and BERT embedding are added to form the input embeddings of the BERT encoder.

4.2 Role Interaction Module

The ERIN we propose uses a Role-Aware Encoder (RAE) to incorporate the role information, and derives contextual features of the input sequence. We donate H as the output of RAE, where $H = \{h_1, h_2, \ldots, h_n, h_{n+1}, \ldots, h_{n+l}\}$, n is the words number of the original input text X and l is the length of text description for the role. We truncate H to obtain the role-specific text representation H_{text} and role representation H_r, where $H_{\text{text}} = \{h_1, h_2, \ldots, h_n\}$ and $H_r = \{h_{n+1}, \ldots, h_{n+l}\}$. A straightforward strategy for event argument extraction is to pass H_{text} into a softmax classifier for predictions. Formally, Y, the set of predictions of arguments are formulated as,

$$Y = \mathrm{softmax}(W_c H_{\text{text}}) \tag{1}$$

where W_c is learnable model parameter and $Y = \{p(\hat{y}_i) | h_i \in H_{\text{text}}\}$, $p(\hat{y}_i)$ is a probability distribution over BIO labels. However, this method ignores the fact that roles of an event are correlated, thus hindering the model from learning explicit interactions between different roles to improve individual argument extraction.

To enhance the correlation between roles, we design a Role Interaction Module (RIM) to dynamically learn explicit interactions between two given roles of an event. The RIM comprises multiple interaction layers, each of which is composed

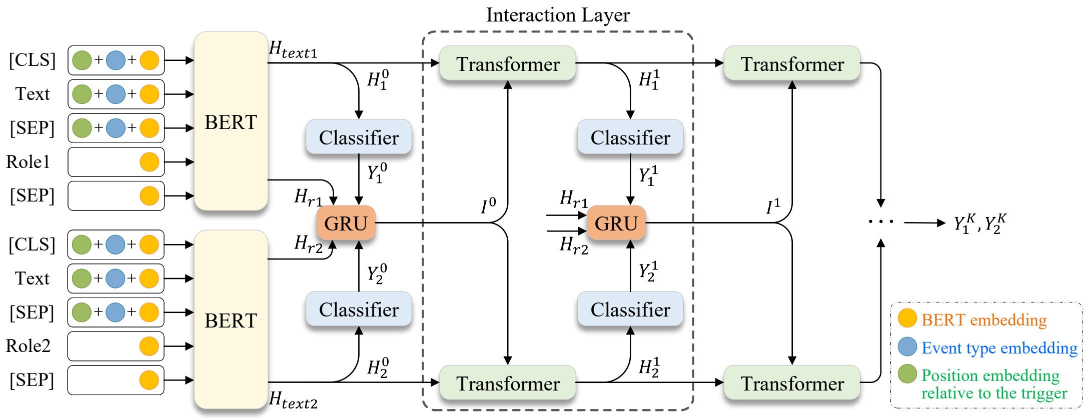

Figure 2: Framework of Explicit Role Interaction Network (ERIN).

of a Transformer (Vaswani et al., 2017), a classifier and a GRU network (Cho et al., 2014). The GRU network is designed to capture the local interaction between outputs of two roles' classifier for argument extraction, establishing a channel for information transmission between two roles, and the Transformer following the GRU network is designed to capture the interaction across roles and enhance role-specific text features.

At the k-th interaction layer, suppose the role-specific text representations and role representations for two given roles r_1,r_2 are H_1^k,H_2^k and H_{r1}^k, H_{r2}^k. Then Y_1^k and Y_2^k, the classifier outputs of argument extraction for r_1 and r_2, are computed as follows,

$$\begin{aligned} Y_1^k &= \text{softmax}(W_c H_1^k) \\ Y_2^k &= \text{softmax}(W_c H_2^k) \end{aligned} \tag{2}$$

The GRU network takes into account the role representations and classifier outputs of argument extraction for r_1 and r_2, and learn an interaction feature I^k. Considering the difference in the correlation between any two roles, we use the reset gate to determine the importance of probability distributions Y_1^k and Y_2^k, and the update gate to control the strength of influence of the roles' correlation on the interaction. The interaction feature I^k at the k-th interaction layer is formally computed as follows,

$$\begin{aligned} z^k &= \sigma(W_z \cdot [H_r, Y^k]) \\ r^k &= \sigma(W_r \cdot [H_r, Y^k]) \\ \tilde{h^k} &= tanh(W_o \cdot [r^k * H_r, Y^k]) \\ I^k &= (1 - z^k) * H_r + z^k * \tilde{h^k} \end{aligned} \tag{3}$$

where $Y^k = [Y_1^k, Y_2^k]$, $H_r = [H_{r1}, H_{r2}]$, $[]$ represents the concatenation operation, $\cdot$ and $*$ represent the matrix-matrix product and element-wise product, σ represents the sigmoid function, and $W = \{W_z, W_r, W_o\}$ are trainable parameters.

To this end, the interaction features I^k derived by the GRU network contains those local interactions between the roles' classifier outputs for argument extraction. Such an interaction is basically word-level. To sufficiently model the interactions across two roles, we employ a Transformer which is primarily composed of a self-attention mechanism, taking as input the interaction features as well as role-specific text representations of a role. The self-attention mechanism allows the inputs to interact with each other, which enables the model to pay attention to the salient features. Its output is an aggregate of these interactions which forms the new role-specific text representations.

Formally, let H_j^{k+1} be the Transformer output at interaction layer k of ERIN, the equations that govern the computation of H_j^{k+1} is given by,

$$\begin{aligned} \hat{H}_j^k &= W_t[H_j^k, I^k] \\ H_j^{k+1} &= \text{softmax}(\frac{W_Q \hat{H}_j^k (W_K \hat{H}_j^k)^T}{\sqrt{d}} W_V \hat{H}_j^k) \end{aligned} \tag{4}$$

where $W = \{W_t, W_Q, W_K, W_V\}$ are trainable model parameters, d is the dimension of text representations, $j \in \{1, 2\}$ specifies the roles r_1 and r_2. To take advantage of the previous learned explicit interactions, we allow ERIN to have a minimum of two interaction layers, i.e, $k = 1, 2, \cdots, K$. The role-specific text representations H_1^K and H_2^K from last interaction layer K are used for final predictions.

4.3 Training Objective

We use the probability distribution Y_1^K and Y_2^K from the last interaction layer as the final predictions of event argument extraction, and perform model training and parameter updating by minimizing the cross-entropy loss. The formula of the training loss is as follows:

$$\text{Loss} = \text{CE}(Y_1^K, \hat{Y}_1) + \text{CE}(Y_2^K, \hat{Y}_2) \quad (5)$$

where $\hat{Y}_j$ is one-hot ground truth of sequence tags and CE is the cross-entropy function.

4.4 Event Detection

Event detection, the upstream task of argument extraction, aims to extract all trigger words t and identify the event types e_t for each extracted t in the input text X. Since event detection is not our main focus in this paper, we develop a simple BERT-based model for this task. Specifically, We employ a pretrained BERT model (Devlin et al., 2019) to encode the input text X and stack a softmax layer on BERT to classify each word into pre-defined event types. The extracted trigger words and its event types are used in the downstream argument extraction task.

5 Experiments

5.1 Dataset

We conduct experiments on the widely used benchmark dataset ACE2005.[3] The dataset covers six forms of resources such as news wire, weblogs, broadcast and speech, and contains 33 event types and 35 argument roles. To ensure a fair comparison, we adopt the same data split as previous work (Sha et al., 2018, 2016; Chen et al., 2015), in which 40 news wire documents were used as the test set, 30 other documents as the development set, and the remaining 529 documents as the training set. Table 1 shows the statistics.

	Train	Dev	Test
Documents	529	40	30
Sentences	14672	873	711
Triggers	4323	492	422
Arguments	7838	931	892

Table 1: Statistics of the ACE2005 dataset.

5.2 Implementation Details

We use a Bert-large-uncased[4] pre-trained model and randomly initialize 1024-dimensional embeddings for evt and pst. Recall, our model considers only two roles in the RIM. Hence, for each role of a given event, we randomly sample another role within the same event to construct the model input.[5] In RIM, the output dimension of GRU and Transformer is set to 512. The two role-aware encoders, all classifiers and GRU networks are parameter-shared, and Transformer is independent for each interaction layer. To alleviate the overfitting problem, we randomly drop 10% neurons in the input layer of the Transformer and classifier. All models are trained with AdamW optimizer (Kingma and Ba, 2015), with a warm-up schedule to smoothen the training process. We perform a grid search to select the best set of hyper-parameters: $K \in [1, 2, ..., 9]$, batch size (bs) $\in [16, 24, 32]$, learning rate $\eta \in [e^{-5}, 2e^{-5}, 3e^{-5}]$ and layers of Transformer (L) $\in [2, 3, 4]$. Best hyper-parameter settings for ERIN are shown in Table 2. All models are implemented with PyTorch library (Paszke et al., 2019) and trained on NVIDIA V100 32GB.

Model	K	η	L	bs
ERIN(Bert-base)	3	$2e^{-5}$	2	24
ERIN(Bert-large)	3	e^{-5}	2	16

Table 2: Best hyper-parameters of ERIN observed on the development set (K: interaction layers, η: learning rate, L: transformer layers, bs: batch size).

5.3 Evaluation Protocol

Following previous work (Li et al., 2013; Yang and Mitchell, 2016), an extracted argument is correct only if the predicted argument span and role exactly matches the gold annotation. Note that if the event detection model wrongly predicts the event type, the extracted arguments for that event will be directly regarded as wrong. We adopt precision (P), recall (R) and micro-f1 (F1) as evaluation metrics. We run the model 5 times with different random seeds and report the average results.

5.4 Performance Comparison

We now compare our model with recent works, including the pipeline-based models: BERT_QA (Du and Cardie, 2020), QA-SL (Zhang et al., 2020) and MQAEE (Li et al., 2020), the joint models: RBPB (Sha et al., 2016), dbRNN (Sha et al., 2018)

[3] https://catalog.ldc.upenn.edu/LDC2006T06

[4] https://github.com/huggingface/transformers

[5] In ACE2005, each event type has an average of 7 roles, and enumerating all role pairs of an event is computationally expensive. Thus, we adopt the random sampling strategy for both training and inference.

Models		ED			EAE		
		P	R	F1	P	R	F1
Pipeline Trigger + Argument	BERT_QA (2020)	71.12	73.70	72.39	56.77	50.24	53.31
	QA-SL (2020)	73.90	68.30	71.00	54.50	52.40	53.40
	MQAEE (2020)	-	-	73.80	-	-	55.00
Joint Trigger + Argument	RBPB (2016)	-	-	67.80	-	-	43.80
	dbRNN (2018)	-	-	69.60	-	-	50.10
	TEXT2EVENT (2021)	69.60	74.40	71.90	52.50	55.20	53.80
Joint Entity + Trigger + Argument	Joint3EE (2019)	68.00	71.80	69.80	52.10	52.10	52.10
	DyGIE++ (2019)	-	-	73.60	-	-	52.50
	JointTransition (2019a)	74.40	73.20	73.80	55.70	51.10	53.30
	GAIL (2019b)	74.80	69.40	72.00	61.60	45.70	52.40
With Additional Supervision	PLMEE (2019)	81.00	80.40	80.70	62.30	54.20	58.00
	RCEE (2020)	75.60	74.20	74.90	-	-	59.30
	RENM (2020)	-	-	73.88	53.50	54.80	55.30
Ours: ERIN	ERIN(Bert-base)	-	-	-	60.27±1.05	53.30±0.30	**56.57±0.49**
	ERIN(Bert-large)	67.55±2.09	74.97±1.91	71.04±1.07	61.70±0.90	54.66±0.89	**57.96±0.77**

Table 3: Event extraction performance comparison with previous models.

and TEXT2EVENT (Lu et al., 2021), the joint models with NER supervision: Joint3EE (Nguyen and Nguyen, 2019), DyGIE++ (Wadden et al., 2019), Joint-Transition (Zhang et al., 2019a), GAIL (Zhang et al., 2019b), and models With addition supervision: PLMEE (Yang et al., 2019), RCEE (Liu et al., 2020) and RENM (Ma et al., 2020), which use external resources or data augmentation strategies to extend the training set. We exclude OneIE (Lin et al., 2020) and GraphIE (Nguyen et al., 2022) in our comparison since they use a different experiment setting, i.e., using 22 role types instead of the 35 role types as used in the compared works.

Table 3 shows the performance comparison between our model and previous work on the test set. We report the results of the entire event extraction process including the two subtasks of event detection and argument extraction, to reflect the performance change for the EAE model only. Since we simply implemented the ED task without any skill, the performance of trigger classification is relatively generic, which is lower than other compared baseline works. The fact that our model ERIN can achieve the SOTA performance on the downstream task even while the upstream task is less powerful is evidence of ERIN's effectiveness.

Viewing the EAE task in further detail, we discover that the NER-integrated approaches (i.e., entity+trigger+argument) outperform RBPB and dbRNN which only extract triggers and arguments, indicating that entity annotations might give the model more supervision. The latest work, TEXT2EVENT, achieves comparable performance to joint entity and event extraction models by uniformly modeling and sharing information across different tasks and labels. Their results suggest that the end-to-end structure makes it easier to learn and exploit correlations between different tasks. In recent years, although joint methods comprise a large proportion, pipeline models have relatively good overall performance as shown in the results. Accordingly, we conduct event extraction in a pipeline manner, where we model the interaction between two arguments and significantly raise the F1 score of SOTA pipeline method MQAEE by 2.96%. PLMEE, RCEE and RENM achieve the best results using cross-domain corpus and data augmentation, indicating the benefits of using sufficient data for the task. Nevertheless, ERIN outperforms RENM without using additional resources and achieves comparable performance to PLMEE.

5.5 Ablation Experiment

To study the contribution of model components, we conduct experiments on the following ablated models: (1) $\text{ERIN}_{\text{w/o GRU}}$: ERIN which replaces the GRU network with a fully-connected network. (2) $\text{ERIN}_{\text{w/o Classifier}}$: ERIN which excludes the classifier in RIM, and removes the argument predictions from the input of GRU network. (3) $\text{ERIN}_{\text{w/o Role}}$: ERIN which excludes the role information when encoding the input text, and uses initial role representations as the input of GRU network. (4) $\text{ERIN}_{\text{w/o Interaction}}$: ERIN which is fed by two identical roles while maintaining the entire network structure and all parameters of RIN. (5) $\text{ERIN}_{\text{w/o RIM}}$: ERIN which excludes the RIM, and directly use the output of RAE for argument extraction. The performance of different ablated models are presented in Table 4.

We find that the performance of ERIN deteriorates as we remove critical components. $\text{ERIN}_{\text{w/o RIM}}$ underperforms ERIN, suggesting the importance of modeling the interaction between

Category	Example	Percentage
Partially Correct	[Three members of a family in India's northeastern state]Target were hacked to death [Attack: Place].	47.0%
Omission	Senior banker John begins [one of the most important jobs in London's financial world]Position when incumbent David steps down [End-Position: Place].	41.2%
Others	—	11.8%

Figure 3: Examples for illustrating different kinds of errors caused by $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on the overlapping arguments. The trigger word is indicates in green, followed by its event type and role of the argument to be extracted. The overlapping arguments provided for interaction are indicated by [bracketed] spans with role types in purple. The predicted arguments by $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ and ERIN are marked by blue and red underlines.

Models	P	R	F1
ERIN	60.27	53.30	56.57
$\mathrm{ERIN}_{\mathrm{w/o\ GRU}}$	60.69	52.21	56.11
$\mathrm{ERIN}_{\mathrm{w/o\ Classifier}}$	58.66	53.47	55.94
$\mathrm{ERIN}_{\mathrm{w/o\ Role}}$	58.66	53.08	55.72
$\mathrm{ERIN}_{\mathrm{w/o\ Interaction}}$	58.29	52.89	55.44
$\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$	58.19	51.92	54.86

Table 4: Performance of different ablated models.

roles for model improvement. Taking a closer look into the components in RIM, we also find $\mathrm{ERIN}_{\mathrm{w/o\ GRU}}$ underperforms relative to ERIN, which implies employing GRU network bring about model improvement. ERIN significantly outperforms $\mathrm{ERIN}_{\mathrm{w/o\ Classifier}}$ especially on the performance of precision, indicating that classifier outputs of argument extraction play an important role in modeling the interaction. We also find that $\mathrm{ERIN}_{\mathrm{w/o\ Role}}$ underperforms ERIN, which is expected since the role representation, encoded together with the input text by pretrained BERT, is more expressive than the initial role embeddings. A further finding is that when excluding the other role information, the F1 score of $\mathrm{ERIN}_{\mathrm{w/o\ Interaction}}$ increases by 0.58% and decreases by 1.13% compared with $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ and ERIN, respectively. It can be concluded that deepening the network (increasing training parameters) can indeed improve the performance, but it is the argument interaction between different roles that plays a greater role.

5.6 Impact on Overlapping Arguments

We now explores the performance of our model on the overlapping arguments. Specifically, We divided the testing instances into non-overlapping and overlapping two subsets, and evaluate ERIN and $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on both subsets.The results are displayed in the Table 5 below.

We find that ERIN consistently improves $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on the instances with or without overlapping arguments. It is not surprised to see the performance improvement on the non-overlapping set, since this performance is consistent with the results discussed above. Interestingly, we find ERIN also significantly surpass $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on the overlapping set. Specifically, ERIN maintains competitive performance with $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on precision while significantly outperforms it on recall. This performance shows that establishing the interaction between two overlapping arguments does not impact the precision but accurately identifies more error-prone arguments that are missed by $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$.

	non-overlapping			overlapping		
	P	R	F1	P	R	F1
w/ RIM	60.15	55.57	57.77	60.66	44.15	51.10
w/o RIM	57.48	53.95	55.64	59.70	41.52	48.96

Table 5: Performance of ERIN and $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ on non-overlapping and overlapping arguments.

To better understand the model behaviour on overlapping arguments, we conduct a further analysis on the overlapping arguments where $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ makes wrong predictions while ERIN makes correct ones, and divided these arguments into 3 categories, as shown in the Figure 3. "Partially Correct" denotes that the predicted argument span of $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ is incomplete, accounting for 47.0% of all items. This type of error might result in inaccurate semantics expressed by arguments. Another major type of error is "Omissions", that is, $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$ fail to identify the arguments. These missing arguments are usually short text spans consisting of 1-3 words, which are difficult to be discovered by $\mathrm{ERIN}_{\mathrm{w/o\ RIM}}$. By informing the model of the span of outer argument with our interaction mechanism, we find ERIN is clever to identify the inner argument. The "Other" categories include those samples for which we cannot find an obvious intuitive pattern.

5.7 Performance on Individual Event Types

We speculate that different event types benefit from role interactions to varying degrees since each event type defines distinct argument roles. We present the performance of ERIN and $\mathrm{ERIN}_{\mathrm{w/o\,RIM}}$ for argument extraction on individual event types in Figure 4.

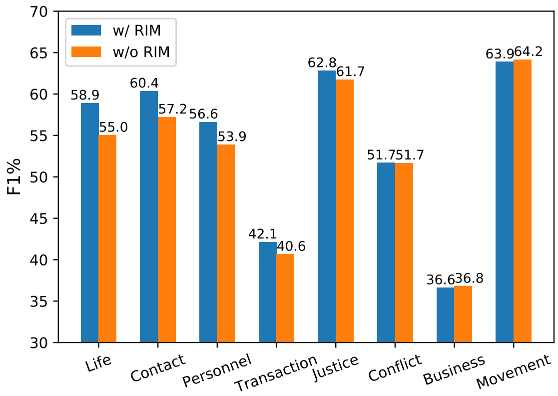

Figure 4: Performance of ERIN and $\mathrm{ERIN}_{\mathrm{w/o\,RIM}}$ on individual event types.

We find that the performance of the first 5 event types has been significantly improved, among which *Life* event has the most improvement (+3.9%), and the ERIN and $\mathrm{ERIN}_{\mathrm{w/o\,RIM}}$ have competitive performances on *Business* and *Movement* events. To better understand the model improvement on the first 5 event types, we then study the arguments that are correctly identified by ERIN but missed by $\mathrm{ERIN}_{\mathrm{w/o\,RIM}}$. Note that for each role of a given event, we have sampled another role within the same event for interaction. Thus, for these arguments, we can pair the role of each argument with the role interacting with it. Interestingly, we find two frequent patterns for these role pairs. The one is that two roles within a pair are homogeneous, accounting for 51% of total role pairs. Homogeneity means two roles belong to the same entity type. For example, both *Victim* and *Agent* in the *Life* event belong to the Person entity type, and *Target* and *Place* in the *Conflict* event belong to the Location type. The semantic similarities between two homogeneous roles may enhance the role information provided for interaction. The other is that the two roles frequently co-occur, accounting for 18%. In the *Justice* event, for example, *Defendant* and *Crime* typically make an appearance together, with *Crime* stating the grounds for which a person is sentenced to be a *Defendant*. Another example is the *Personnel* event, where the *Position* role has clear semantics only when *Entity* role (i.e., company) is present. This pattern implies our model is clever to leverage the co-occurrence relationship between roles to promote the argument extraction.

5.8 Impact of Interaction Layers K

Figure 5 illustrates the model's F1 curve on the test set given different values of the hyper-parameter K. The goal is to show the impact of the number of interaction layers on model performance. We observe that ERIN is reduced to the ablation model "w/o RIM" at $K = 0$. We see that the model's performance improves significantly as K increases, with F1 growing at a rapid rate and peaking at $K = 3$. Afterwards, F1 oscillates and gradually decays because the model starts to over-fit.

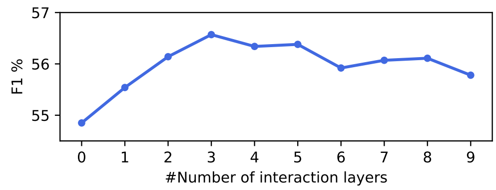

Figure 5: Curves of F1 on different number of interaction layers K.

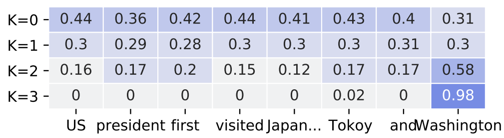

Figure 6: Probability visualization on words predicted as *Entity* with 0, 1, 2 and 3 layers of interaction.

Figure 6 shows a probability visualization of our case example illustrated as our motivation in Figure 1(a). Here, we observe the probabilities assigned to words at different values of $K \in [0, 1, 2, 3]$ to detect the *Entity* argument. At $K = 0$, there is no distinction between the predicted probabilities among words, indicating that the model is confused about identifying the arguments in the text when the interaction is not modelled. As K increases, we observe that the correct *Entity* "Washington" is detected with a probability of 0.98. ERIN achieves this by gradually taking advantage of interactions among words in higher interaction layers.

6 Conclusion and Future Work

In this paper, we propose a novel model for the task of argument extraction, referred to as Explicit

Role Interaction Network, to capture the correlation between different argument roles within the same event. The model is able to generate role-aware text representations and explicitly create role-information transmission between the predicted positions of two arguments in the sequence using the GRU network, enabling argument spans to be mutually aware. This structure not only aids in the detection of overlapping arguments but also dynamically adjusts and refines the identification results based on the multi-layer interaction. Extensive experiments on ACE 2005 demonstrate the effectiveness of this method. In future, we intend to use the SOTA model of event detection (Li et al., 2021) to obtain reliable predicted triggers as the input for argument extraction, to explore the upper bound performance of our proposed method. In addition, we will extend the explicit interaction method to other IE tasks, such as document event extraction and relation extraction, to study its application scope.

Limitations

Our model stacks multiple layers of Transformer after the pretrained model, which causes an increase in training parameters and slower model convergence. According to our statistics, besides the parameters of BERT itself, ERIN(Bert-base) and ERIN(Bert-large) include additional training parameters of 3.2 and 19.1 millions, and the time spent on each training epoch on the dataset is 3.6 and 12 minutes, respectively. Since the model requires dozens of training epochs and there are many hyper-parameters to tune, it is necessary to speed up the training process for faster development of the final model. Another limitation is that we do not take the co-reference problem into account. In a small set of ACE2005, the same entity may refer to different description spans in the text. This issue should be further investigated for both effective modeling and accurate evaluation. We consider that this problem can be alleviated by leveraging syntax information and knowledge of the corpus.

Acknowledgements

This work was supported in part by the National Key R&D Program of China under Grant 2021ZD0110700, in part by the Fundamental Research Funds for the Central Universities, in part by the State Key Laboratory of Software Development Environment. SM is supported by a Leverhulme Trust Research Project Grant (No. RPG-2020-148).

References

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 of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 167–176. The Association for Computer Linguistics.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1724–1734. ACL.

Shiyao Cui, Bowen Yu, Tingwen Liu, Zhenyu Zhang, Xuebin Wang, and Jinqiao Shi. 2020. Edge-enhanced graph convolution networks for event detection with syntactic relation. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 2329–2339. 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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 671–683. Association for Computational Linguistics.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 504–515. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations,* *ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Viet Dac Lai, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2020. Event detection: Gate diversity and syntactic importance scores for graph convolution neural networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 5405–5411. Association for Computational Linguistics.

Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 829–838. Association for Computational Linguistics.

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, ACL 2013, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, pages 73–82. The Association for Computer Linguistics.

Rui Li, Wenlin Zhao, Cheng Yang, and Sen Su. 2021. Treasures outside contexts: Improving event detection via global statistics. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021*, pages 2625–2635. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, pages 7999–8009. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 1641–1651. Association for Computational Linguistics.

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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2795–2806. Association for Computational Linguistics.

Jie Ma, Shuai Wang, Rishita Anubhai, Miguel Ballesteros, and Yaser Al-Onaizan. 2020. Resource-enhanced neural model for event argument extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020, Online Event, 16-20 November 2020*, volume EMNLP 2020 of *Findings of ACL*, pages 3554–3559. Association for Computational Linguistics.

Minh Van Nguyen, Bonan Min, Franck Dernoncourt, and Thien Huu Nguyen. 2022. Joint extraction of entities, relations, and events via modeling inter-instance and inter-label dependencies. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 4363–4374. Association for Computational Linguistics.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6851–6858. AAAI Press.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. Pytorch: An imperative style, high-performance deep learning library.

Lei Sha, Jing Liu, Chin-Yew Lin, Sujian Li, Baobao Chang, and Zhifang Sui. 2016. RBPB: regularization-based pattern balancing method for event extraction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge RNN and tensor-based argument interaction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5916–5923. AAAI Press.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2020. Recurrent interaction network for jointly extracting entities and classifying relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages

3722–3732. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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 2019, Hong Kong, China, November 3-7, 2019*, pages 5783–5788. Association for Computational Linguistics.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 289–299. The 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 Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 5284–5294. Association for Computational Linguistics.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019a. Extracting entities and events as a single task using a transition-based neural model. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5422–5428. ijcai.org.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019b. Joint entity and event extraction with generative adversarial imitation learning. *Data Intell.*, 1(2):99–120.

Yunyan Zhang, Guangluan Xu, Yang Wang, Daoyu Lin, Feng Li, Chenglong Wu, Jingyuan Zhang, and Tinglei Huang. 2020. A question answering-based framework for one-step event argument extraction. *IEEE Access*, 8:65420–65431.

Sendong Zhao, Ting Liu, Sicheng Zhao, and Fei Wang. 2019. A neural multi-task learning framework to jointly model medical named entity recognition and normalization. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 817–824. AAAI Press.