Mining Information from Event Structure Relation Graph for Event Argument Extraction

Anonymous ACL submission

Abstract

Event Argument Extraction is a vital subtask of Event Extraction. Despite the achievements in existing methods, they can not fully use the event structure information and the rich semantics of the labels, which can provide richer external knowledge for extracting event arguments. To this end, we propose an efficient and end-to-end event argument extraction model based on the **E**vent **S**tructure and **Q**uestion **A**nswering (**ESQA-EAE**): (1) we model a multi-relational graph of event ontologies to get the structure-aware node representations; (2) we encode the questions and event mentions separately to avoid premature fusion of the two features. Experiments on the ACE2005[1] show that ESQA-EAE surpasses the baseline models, which further show that ESQA-EAE can use the structural information to improve the accuracy of event argument extraction.

1 Introduction

Event Argument Extraction (EAE) aims to identify the event arguments and classify their roles in the event mention, according to the given event type and trigger word. As in the sentence "Tugle was on trial for raping and killing a southwest Virginia grandmother.", the event type is *Conflict.Attack* triggered by the word "raping", EAE needs to extract "Tugle" and "grandmother" as two arguments, and classify their roles into *Attacker* and *Target* respectively.

Most of the existing methods regard EAE as an entity classification task (Chen et al., 2015; Nguyen et al., 2016; Liu et al., 2018; Sha et al., 2018;Wang et al., 2019; Ma et al., 2020; Xiangyu et al., 2021; Ahmad et al., 2021), a sequence labeling task (Ma et al., 2020; Yang et al., 2018; Chen et al., 2020a; Du and Cardie, 2020a), or or a joint learning task (Nguyen and Nguyen, 2019; Wadden et al., 2019; Lin et al., 2020). Recently, some studies model it as a Machine Reading Comprehension (MRC) / Question Answering (QA) task(Yang et al., 2019; Du and Cardie, 2020b; Liu et al., 2020; Li et al., 2020; Chen et al., 2020b; Zhou et al., 2021; Zhang et al., 2020), which can solve the shortcomings in the previous methods well. Yet, there are still weaknesses in the existing methods: (1) The existing methods cannot make full use of the complex relations between events and argument roles; (2) MRC / QA methods need to design the questions carefully; (3) MRC / QA methods encode the questions and the contexts jointly, fusing the information of the two prematurely.

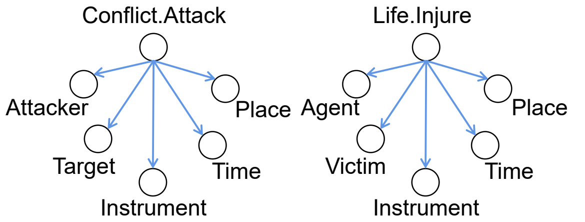

Figure 1: Event Type Structures of *Conflict.Attack* and *Life.Injure*. The event types and argument roles are defined in ACE2005, where argument roles *Instrument*, *Time* and *Place* are in both two event type structures.

For tackling the weaknesses, we propose an efficient and end-to-end event argument extraction model based on the **E**vent **S**tructure and **Q**uestion **A**nswering (**ESQA-EAE**). For weaknesses (1), we assume that there are complex relations between event types and argument roles, which can be used as external knowledge for EAE. Huang et al. (2018) pointed out that Event Ontology can be represented by structure, which defines each event type and a set of argument roles and the relation between them. As shown in Fig. 1, we found that there are common roles in different event type structures, which implies certain information. To better utilize this information, we model the event type structures as a

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

multi-relational graph to obtain the structure-aware node representations for event argument QA.

For weakness (2), we assume what works in a question are the keywords, and the other words in the well-designed query will introduce noise and cause unnecessary encoding costs. For weakness (3), we assume that fully understanding the question and context is the key to answering correct answers, while jointly encoding causes attention distraction. Thus, ESQA-EAE encodes the question and event mention separately, and then the fusion features are used for predicting answers. Note that in ESQA-EAE, only the event types and the argument roles are used to construct questions, simplifying the question design. Besides, ESQA-EAE makes it possible to extract multiple argument answers simultaneously to overcome the multi-arguments problem.

In general, the contributions of this paper are:

• We model a multi-relational graph ESRG for the complex relations in event type structures, which is used as external knowledge for event argument QA.

• We propose a model ESRG-EAE that encodes the questions and event mentions separately and make use of the features learned from ESRG as questions for event argument extraction.

• Experiment results show that our proposed model outperforms the baseline models.

2 Related Works

2.1 Event Argument Extraction

The existing event argument extraction studies can be divided into the following categories.

2.1.1 Entity Classification task

Most researchers model EAE as an entity classification task, that is, classify the corresponding argument roles for the candidate argument entities. Chen et al. (2015) introduce dynamic multi-pooling layer to reserve more crucial information. Nguyen et al. (2016) use RNN and memory matrices. Liu et al. (2018) make use of semantic arcs and graph attention convolution. Sha et al. (2018) introduces syntactic dependency bridge into RNN. Wang et al. (2019) proposes "superordinate concept" and use the concept hierarchy for EAE. Ma et al. (2020) introduce a Syntax-Attending Transformer. Xiangyu et al. (2021) propose a novel Bi-directional Entity-level Recurrent Decoder. Ahmad et al. (2021) proposes Graph Attention Transformer Encoder that it takes into account the syntactic structure and distances.

2.1.2 Joint Method

The error propagation problem between Named Entity Recognition(NER) and EAE has prompted scholars to study joint methods of the two tasks. One is to model EAE as a sequence labeling task(Ma et al., 2020; Yang et al., 2018; Chen et al., 2020a; Du and Cardie, 2020a), that is, classify a BIO label[2] for each word, and obtain one optimal extraction in combination with Conditional Random Field(CRF) and Viterbi Algorithm, therefore the NER is not required. Another approach is to jointly learn the NER and EAE in one model(Nguyen and Nguyen, 2019; Wadden et al., 2019; Lin et al., 2020), for alleviating the error propagation problem.

2.1.3 Machine Reading Comprehension / Question Answering

Recently, some studies model the EAE as a Machine Reading Comprehension(MRC) / Question Answering(QA) task. The model needs to understand the context(i.e., event mention) and answer the questions related to argument roles, in which the answers are the event arguments. Yang et al. (2019) use BERT(Devlin et al., 2019) as the feature extractor and extract arguments based on roles. Du and Cardie (2020b) and Zhang et al. (2020) generate questions for each role. Liu et al. (2020) use templates and unsupervised style transfer model to construct questions. Li et al. (2020) models EAE as multi-turns QA task. Chen et al. (2020b) requires the model to fill the extraction templates. Zhou et al. (2021) proposes a semi-supervised EAE approach via Dual Question Answering.

2.2 Graph

Graph can easily model complex relations, which has attracted many researchers(Kipf and Welling, 2017; Veličković et al., 2018; Schlichtkrull et al., 2018; Wang et al., 2020; He et al., 2020; Fu et al., 2019; Zeng et al., 2020). Some of EAE researches also make use of Graph information, Liu et al. (2018) uses graph attention convolution to aggregate the syntactic information. Wadden et al. (2019) does information extraction using dynamically constructed span graphs. Ahmad et al. (2021) intro-

[2] "B-":the begining of an argument, "I-": inside of an argument, "O": not a part of an argument.

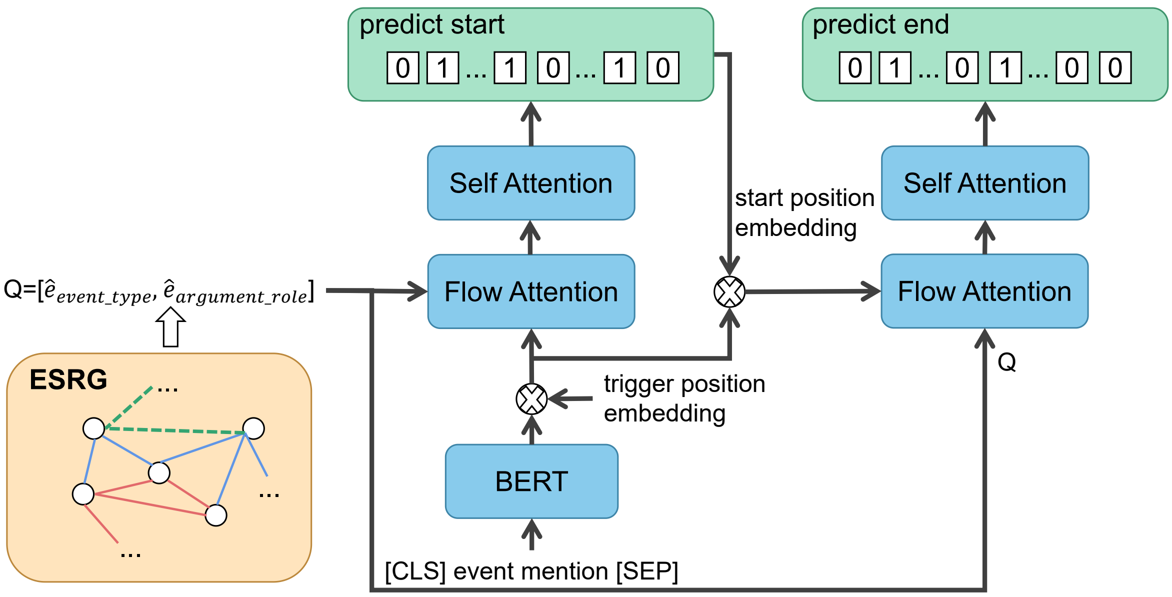

Figure 2: Model architecture of ESQA-EAE

duces the syntactic distances in the syntactic structure into Transformer.

3 Methodology

3.1 task setup

Consider an event mention sentence $EM = \{w_1, .., w_{tri}, .., w_n\}$ with n tokens, w_{tri} is the trigger of the event type $event\_type_t$, where $event\_type_t$ belongs to a fixed set of pre-defined event types. Given w_{tri} and $event\_type_t$, EAE aims to identify all argument spans from EM and classify the role r for each argument, where r belongs to a fixed set of pre-defined roles for the $event\_type_t$. An extracted argument can be expressed as $[s, e, r]$, where s / e is the index of the start / end token of the argument in the EM.

Fig.2 presents our model architecture, which will be explained in detail in the following subsections.

3.2 Event Structure Relation Graph (ESRG)

We assume that there are complex relations between event types and argument roles. To capture these features, we use event ontologies to model a multi-relational graph and encode the graph using Attention Mechanism to obtain the structure-aware node representations.

3.2.1 Graph building

We connect event type structures and expand the relational connections, and model as a multi-relational undirected graph, denoted as **E**vent **S**tructure **R**elation **G**raph (**ESRG**) $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathcal{R}\}$, $\mathcal{V}$ is composed of event type nodes and argument roles nodes[3], $\mathcal{E}$ is the set of edges, $\mathcal{R} = \{r_1, r_2, r_3\}$ is the set of three relations types, where $(\nu_i, r_k, \nu_j) \in \mathcal{E}$.

Fig.3 shows an ESRG with only two event types: *Life.Die* and *Conflict.Attack*. There are three types of relations in an ESRG:

- **Event Type-Event Type**, that is, any two event type nodes are connected to capture the correlation and dependency between the events.
- **Event Type-Argument Role**, the event type node and its corresponding set of argument role nodes are connected to capture the structural information within the event type structure.
- **Argument Role-Argument Role**, any two role nodes are connected to capture the similarities and correlations between the roles.

We explicitly model the complex relations into an ESRG, update the features of nodes so that each node aggregates the features of the neighbors propagated from three kinds of relations to obtain the structured-aware node representations.

3.2.2 Node feature initialization

We use BERT (Devlin et al., 2019) as the feature extractor for initializing each node's feature, each node ν_i should be processed into a standard BERT-style format "$[CLS]\nu_i[SEP]$"[4] as input, then we take the output of the $[CLS]$ of the last hidden layer

[3] We have conducted experiments to distinguish these two types of nodes, which did not significantly affect the results.

[4] $[CLS]$ and $[SEP]$ are two special tokens in BERT.

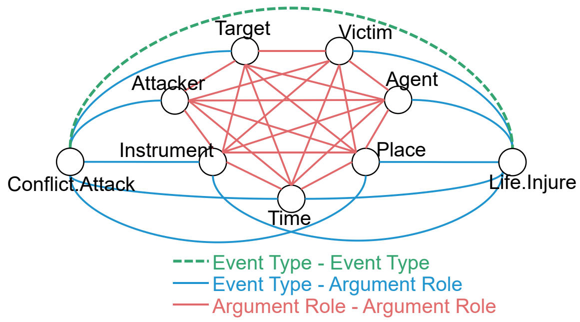

Figure 3: Event Structure Relation Graph for only two event types. *Conflict.Attack* and *Life.Injure* are two event type nodes, and the rest are all argument role nodes. There are three type of edges between nodes.

as the initial feature e_i of the node ν_i:

$$e_i = BERT_{[CLS]}([CLS]\nu_i[SEP]) \tag{1}$$

Therefore we get the initial feature matrix of all nodes $E = \{e_1, ..., e_m\} \in \mathbb{R}^{d*m}$, where d is the dimension of the BERT's last hidden layer and m is the number of nodes.

3.2.3 Relational Graph Attention Network (RGAT)

Inspired by RGCN (Schlichtkrull et al., 2018) and R-GAT (Wang et al., 2020), we adopt Attention Mechanism into RGCN to learn the Event Structure Relation Graph.

Specifically, in the L-th RGAT layer, we first calculate the correlation score s_{ij}^L of any two different nodes ν_i and ν_j, the scoring function we adopted is proposed by (Luong et al., 2015):

$$s_{ij}^L = score(e_i^L, e_j^L) \tag{2}$$

$$score(e_i^L, e_j^L) = \sigma(W_s^L[e_i^L || e_j^L]) \tag{3}$$

where W_s^L is learnable weight matrix, $\sigma(\cdot)$ is the activation function, "||" denotes the concatenation, and $e_i^0 = e_i$. Then, the relational adjacency matrix is used as the attention mask to calculate the attention weight $\alpha_{r,ij}^L$ between nodes ν_i and ν_j under specific relation $r \in \mathcal{R}$:

$$\alpha_{r,ij}^L = \frac{exp(s_{ij}^L)}{\sum\limits_{k \in N_i^r} exp(s_{ik}^L)} \quad (j \in N_i^r) \tag{4}$$

where N_i^r is the neighbor set of node ν_i under relation $r \in R$. The attention weight is used to aggregate the features of neighbors, thus the feature update for node ν_i at L-th RGAT layer can be formulated as:

$$e_i^{L+1} = \sigma(\sum_{r \in R} \sum_{j \in N_i^r} \frac{1}{c_i^r} \alpha_{r,ij}^L W_r^L e_j^L + \alpha_{ii}^L W_0^L e_i^L) \tag{5}$$

where c_i^r is a normalization constant, W_r^L and W_0^L are learnable weight matrices.

Through the learning of L layers RGAT, we obtain the structure-aware node representations, $\tilde{E} = \{\tilde{e}_1, ...\tilde{e}_m\} \in \mathbb{R}^{d*m}$. In Section 3.3.2, we will explain how to construct questions for EAE based on the features of the nodes.

3.3 Event Argument Extraction based on Question Answering

We assume that: (1) what really works in the question are the keywords; (2) fully understanding the question and context is the key to answering correct answers. Therefore, we propose a Question Answering method that encodes the questions and event mentions separately, and the fusion features of them are used for predicting answers.

3.3.1 Event Mention Encoding

Same as Section 3.2.2, we employ BERT(Devlin et al., 2019) to encode the EM. Firstly convert the EM into the input format of BERT, then take the output of BERT's last hidden layer as the initial representation of the event mention H^{EM}:

$$\begin{aligned} H^{EM} &= BERT([CLS]EM[SEP]) \\ &= BERT([CLS]w_1, .., w_{tri}, .., w_n[SEP]) \end{aligned} \tag{6}$$

where $H^{EM} \in \mathbb{R}^{d*N}$, N is the length of the BERT input. Besides, we introduce two embeddings to jointly construct event mention representation for EAE:

• **Trigger Position Embedding,** E^{tri} : EAE depends on the event type determined by the trigger word, so the model needs to know which is the trigger word (Yang et al., 2019). Thus, we introduce a learnable embedding to every token indicating whether it is the trigger word or not, named Trigger Position Embedding.

• **Start Position Embedding,** E^{start} : When predicting the end of argument spans, we introduce the learnable Start Position Embedding to every token indicating whether it is a start of an argument or not, flowing information from the start predictor to the end predictor.

So, the event mention representation for predicting the start of answers H_s^{EM} is composed of

H^{EM} and E^{tri}, while H_e^{EM} for predicting the end of answers is composed of H^{EM}, E^{tri} and E^{start}:

$$\begin{aligned} H_s^{EM} &= \sigma(W_s[H^{EM}||E^{tri}] + b_s) \\ H_e^{EM} &= \sigma(W_e[H^{EM}||E^{tri}||E^{start}] + b_e) \end{aligned} \tag{7}$$

where $H_s^{EM}, H_e^{EM} \in \mathbb{R}^{d*N}$, W_s, b_s, W_e and b_e are learnable parameters.

3.3.2 Question Constructing

The question we designed only needs two elements: event type and argument role, and utilize the features of the nodes obtained in Section 3.2 to construct the question representation:

$$\begin{aligned} Q &= \{q_1, q_2\} \\ &= \{\hat{e}_{event\_type}, \hat{e}_{role}\} \end{aligned} \tag{8}$$

where $\hat{e}_{event\_type}$ and $\hat{e}_{role}$ are representations of the event type and the argument role, fused by the initial feature in Section 3.2.2 and the updated feature in Section 3.2.3 of the corresponding node:

$$\begin{aligned} \hat{e}_\theta =& Fusion(e_\theta, \tilde{e}_\theta) \\ & \theta \in \{event\_type, role\} \end{aligned} \tag{9}$$

where $Fusion(\cdot)$ is the feature fusion function, which can be summation, averaging or concatenation, $Q \in \mathbb{R}^{d*2}$. Thus, $\hat{e}_\theta$ is a rich semantic representation that integrates the semantics of the label itself and the relational structure information, which is crucial to answering questions correctly.

3.3.3 Flow Attention

To combine the features of the question and the context, we follow (Zhou et al., 2021) and (Seo et al., 2016) to use Flow Attention to generate question-aware context representation:

$$H_\xi^{FA} = FlowAtt(H_\xi^{EM}, Q)\ (\xi \in \{s, e\}) \tag{10}$$

where $FlowAtt(\cdot)$ is the Flow Attention function. It takes the event mention representation and question representation as input, and outputs the question-aware event mention representation we wanted.

Flow attention calculates the attention between question and event mention from two directions: from event mention to question (EM2Q) and from question to event mention (Q2EM). Firstly, the similarity matrix of event mention and question is calculated:

$$\begin{aligned} SA_\xi =& \delta(H_\xi^{EM}, Q) \\ =& MLP([H_\xi^{EM}||Q||(H_\xi^{EM} \circ Q)]) \\ & (\xi \in \{s, e\}) \end{aligned} \tag{11}$$

where $MLP(\cdot)$ is a Multilayer Perceptron, "$\circ$" denotes element-wise multiplication, $SA_{\xi,ij}$ indicates the similarity between the i-th token in the EM and the jth element in Q while predicting start / end. Then it use SA_ξ to calculate attention from two directions.

EM2Q Indicates the most relevant question element (event type or argument role) to each event mention token. First calculates the EM2Q attention score $\eta_{\xi,ij}$, and then aggregate the features from EM to Q according to $\eta_{\xi,ij}$ to produce the feature vector h_ξ^{EM2Q}:

$$\begin{aligned} \eta_{\xi,ij} &= \frac{exp(SA_{\xi,ij})}{\sum_{k=1}^{|Q|} exp(SA_{\xi,ik})} \\ h_{\xi,i}^{EM2Q} &= \sum_j \eta_{\xi,ij} q_j \\ & (\xi \in \{s, e\}) \end{aligned} \tag{12}$$

where q_j indicates the jth feature vector of Q. Therefore, we get the EM2Q event mention representation $H_\xi^{EM2Q} = \{h_{\xi,1}^{EM2Q}, ..., h_{\xi,N}^{EM2Q}\} \in \mathbb{R}^{d*N}$ $(\xi \in \{s, e\})$.

Q2EM Indicates which event mention tokens have the closest similarity to one of the question elements, which are very important for answering. First calculates the Q2EM attention score $\mu_{\xi,ij}$, and then aggregate the features to produce h_ξ^{Q2EM}:

$$\begin{aligned} \mu_{\xi,i} &= \frac{exp(max(SA_{\xi,i:}))}{\sum_{k=1}^{N} exp(max(SA_{\xi,k:}))} \\ h_\xi^{Q2EM} &= \sum_i \mu_{\xi,i} h_{\xi,i}^{EM} \\ & (\xi \in \{s, e\}) \end{aligned} \tag{13}$$

where $SA_{k:}$ denotes the i-th row elements of SA, $h_{\xi,i}^{EM}$ is the i-th feature vector of H_ξ^{EM} $(\xi \in \{s, e\})$. Therefore, h_ξ^{Q2EM} is the weighted sum of the most important features in the event mention about the question, and it is tiled N times to form the Q2EM event mention representation, that is $H_\xi^{Q2EM} \in \mathbb{R}^{d*N}$ $(\xi \in \{s, e\})$.

Finally, H_ξ^{EM2Q} and H_ξ^{Q2EM} are combined together to yield $H_\xi^{FA} \in \mathbb{R}^{4d*N}$ as output:

$$\begin{aligned} H_\xi^{FA} =& \beta(H_\xi^{EM}, H_\xi^{EM2Q}, H_\xi^{Q2EM}) \\ =& [H_\xi^{EM} || H_\xi^{EM2Q} || (H_\xi^{EM} \circ H_\xi^{EM2Q}) \\ & || (H_\xi^{EM} \circ H_\xi^{Q2EM})] \\ & (\xi \in \{s, e\}) \end{aligned} \tag{14}$$

3.3.4 Self Attention

To further integrate the features of event mention and question, we add a Self Attention Layer after the Flow Attention Layer to obtain the event mention representation for predicting answer spans.

$$\begin{aligned} H_\xi^{SA} = & Attention(H_\xi^{FA}, H_\xi^{FA}, H_\xi^{FA}) \\ & (\xi \in \{s, e\}) \end{aligned} \tag{15}$$

Where $Attention(\cdot)$ is the attention function, we implement it with reference to (Vaswani et al., 2017) and (Bahdanau et al., 2015).

$$\begin{aligned} & Attention(Q, K, V) \\ & = softmax(W_a^T tanh(W_q Q + W_k K))V \end{aligned} \tag{16}$$

where W_a, U_q and W_k are learnable parameters.

3.3.5 Prediction

Instead of using two N-classifiers to predict the start and end of an answer, which can not solve the multi-answers problem, we adopt N 2-classifiers to predict the probability of whether each token in the event mention is the start / end of an answer.

$$start\_prob = sigmoid(W_{sp} H_s^{FA} + b_{sp}) \tag{17}$$

$$end\_prob = sigmoid(W_{ep} H_e^{FA} + b_{ep}) \tag{18}$$

where W_{sp}, b_{sp}, W_{ep} and b_{ep} are learnable parameters.

Finally, we run the Answer Span Matching Algorithm (ASMA) to obtain all the extracted answer spans as event arguments. Specifically, ASMA first finds the index s that is inside the event mention and $start\_prob[s] \geq start\_threshold$ as a start of an answer. Next to find the index e that nearest s, inside the event mention and $end\_prob[e] \geq end\_threshold$, as the end. Add the answer $[s, e, event\_type, role]$ to the $answer\_list$, and repeat the above steps until the appropriate start index cannot be found.

3.4 Loss Function

We adopt Binary Cross-Entropy for calculating the loss between the predicted result and ground truth. The final loss $Loss$ is the sum of the start token loss $Loss_s$ and the end token loss $Loss_e$:

$$Loss = Loss_s + Loss_e \tag{19}$$

$$Loss_s = BCE(start\_prob, start\_label) \tag{20}$$

$$Loss_e = BCE(end\_prob, end\_label) \tag{21}$$

4 Experiments

4.1 Experimental Setup

Dataset and Evaluation We conduct experiments on the dataset ACE2005 (Walker et al., 2006), which annotated 33 event subtypes and 35 argument roles. For a fair comparison with other methods, we use the same data split and preprocessing step as in the prior works (Wadden et al., 2019; Du and Cardie, 2020b), retaining 33 event subtypes and 22 roles.

We also adopt the same criteria they used: (1)An event argument is correctly identified if the start and end offset and the event type match those of any of the arguments labeled (**AI**); (2)It is correctly classified if the argument role is also correct (**RC**). The criteria mentioned above are evaluated using Precision (P), Recall (R), and F1 score (F1).

4.2 Experiment Details

We utilize BERT-Base as the feature extractor, which has 12 layers, 768 hidden units, and 12 attention heads. AdamW is used as the optimizer with the learning rate is 5e-5, and the weight decay is 0.01. The embedding size of Trigger Position Embedding and Start Position Embedding is 50. We use 2-layer RGAT to update the features of the nodes in ESRG. In ASMA, we limit the max length of an answer span to 5.

4.3 Baseline Models

We compare our model with: (1) **dbRNN** (Sha et al., 2018) leverages the dependency information; (2) **Joint3EE** (Nguyen and Nguyen, 2019) proposes a model to perform predictions for entities and events jointly; (3) **DyGIE++** (Wadden et al., 2019), a framework that models the spans and captures within-sentence and cross-sentence context; (4) **GAIL-ELMo** (Zhang et al., 2019), an ELMo-based inverse reinforcement learning method using a generative adversarial network(GAN) for entity and event extraction; (5) **EEQA** (Du and Cardie,

Model	Argument Identification(AI)			Role Classification(RC)		
	P	R	F1	P	R	F1
dbRNN	-	-	57.20	-	-	50.10
Joint3EE	59.90	59.80	**59.90**	52.10	52.10	52.10
DyGIE++	-	-	55.40	-	-	52.50
GAIL-ELMo	**63.60**	48.70	55.10	**61.60**	45.70	52.40
EEQA	58.90	52.80	55.29	56.77	50.24	53.31
(Ma et al., 2020)	58.40	56.90	57.60	56.00	54.80	55.30
ESQA-EAE	53.51	**62.15**	57.51	51.72	**60.07**	**55.58**

Table 1: Overall Result on ACE2005

Model	Argument Identification			Argument Role Classification		
	P	R	F1	P	R	F1
EEQA	**58.90**	52.80	55.29	**56.77**	50.24	53.31
ESQA-EAE	53.51	62.15	**57.51**	51.72	**60.07**	**55.58**
w/o ESRG	51.61	**64.06**	57.16	48.39	60.07	53.60

Table 2: Results of Ablation Study

2020b) formulates event extraction as a QA task; (6) Ma et al. (2020) introduces a syntax-attending Transformer for event argument extraction.

4.4 Overall Result

Table 1 shows the comparison between ESQA-EAE and baseline models[5]. We observed that: (1) ESQA-EAE achieves the best Recall and F1 score on RC; (2) ESQA-EAE can make up for the gap in event detection, which shows that ESQA-EAE is less sensitive to the results of event detection and focuses more on the event argument extraction itself; (3) ESQA-EAE only takes event types and argument roles as questions, and the result shows that ESQA-EAE is significantly better than EEQA, which confirms our hypothesis: what really works in the question are the keywords and fully understanding is the key to answering correct answers.

4.5 Ablation Study

To better understand the effectiveness of the Event Structure Relation Graph we proposed, we ablate the ESRG. We (w/o ESRG) only construct the questions based on the initial features. The results are shown in Table 2: (1) All indices on RC of the ablation model are significantly decreased, which proves our hypothesis: there are complex relations in the event structure, which can provide richer external knowledge for extracting event arguments. (2) In addition, the ablation model can still exceed the performance of EEQA. This observation proves another hypothesis: fully understanding the semantics of questions and contexts is the key to answering correct answers, and separate encoding can avoid premature integration of the features of questions and contexts.

[5] Although event detection is not the focus of our work, for a fair comparison with other methods, we adopt an event detection QA same as EEQA(Du and Cardie, 2020b) to generate the event detection results to test the effect of ESQA-EAE. In our experiment, the trigger classification F1 score is **71.03**. Note that this result will directly affect the performance of EAE.

4.6 Complex Data Scenarios

To further explore the performance of our proposed model in complex data scenarios, we build different subsets of the test set according to the special scenario for testing:

• **Multi-Arguments / Multi-Answers (MA)** We construct a subset of data with more than two event arguments in an argument role. One sample is identified by the event mention, event type, and role. The constructed subset contains 54 samples. We only report the F1 score on RC under the golden triggers.

• **No-Arguments / No-Answers (NA)** We construct a subset of data with no event argument in an event. One sample is identified by the event mention and the event type. The constructed subset contains 91 samples. Since there are no golden

Model	Data Scenarios		
	Multi-Arguments(MA) (F1 on RC)	No-Arguments(NA) (wrong answers count)	Multi-Events(ME) (F1 on RC)
EEQA†	65.55	65	67.23
ESQA-EAE	**76.79**	**33**	**72.13**

Table 3: Results on different data scenarios

arguments for calculating PRF value, we report the number of extracted answers, which are all wrong.

• **Multi-Events (ME)** We construct a subset of data with multiple events in an event mention. Each event may fit the previously mentioned data scenario, making it more complex than the scenarios mentioned above. An event mention identifies one sample, and the subset contains 99 samples. We report the F1 score on RC under the golden triggers.

We compare our model with EEQA†[6], our model is consistent with the one reported in Section 4.4. As shown in table 3, ESQA-EAE outperforms EEQA† in all scenarios: (1) In the MA scenario, ESQA-EAE surpasses the F1 score of EEQA† significantly. EEQA transforms multi-answer extraction into multiple rounds of QA with different standard answers during training, while at test time, the model has to predict multiple answers, which confuses the model. In contrast, ESQA-EAE extracts multiple answers simultaneously in one turn QA, keeping consistent during training and testing. (2) In the NA scenario, the number of incorrect answers extracted by ESQA-EAE is significantly less than EEQA†. This observation indicates that ESQA-EAE can better capture the global information via fully knowing the context itself for judging whether there is an argument / answer (similar to the global feature in OneIE (Lin et al., 2020)). (3) ESQA-EAE achieves a better F1 score in the ME scenario. By modeling the event structure, ESQA-EAE takes the complex relations to guide the EAE. Besides, ESQA-EAE independently extracts the semantic representation of the event mention to avoid fusing information of querys in advance, We argue that this strategy is closer to human reading, i.e., the semantics of the context itself should not change due to the questions.

[6]EEQA† is obtained by re-implementing EEQA(Du and Cardie, 2020b) using the best question generation strategy, the best F1 score we got is **64.43**.

5 Conclusion

We propose an efficient and end-to-end event argument extraction model ESQA-EAE, which utilizes the structure information to guide the event argument question answering. ESQA-EAE simplifies the previous QA methods and achieves better scores. The limitations on few-shot learning and document-level extraction will be our future works.

References

Wasi Uddin Ahmad, Nanyun Peng, and Kai-Wei Chang. 2021. Gate: Graph attention transformer encoder for cross-lingual relation and event extraction. In *The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Pei Chen, Hang Yang, Kang Liu, Ruihong Huang, Yubo Chen, Taifeng Wang, and Jun Zhao. 2020a. Reconstructing event regions for event extraction via graph attention networks. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 811–820, Suzhou, China. 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.

Yunmo Chen, Tongfei Chen, Seth Ebner, Aaron Steven White, and Benjamin Van Durme. 2020b. Reading the manual: Event extraction as definition comprehension. In *Proceedings of the Fourth Workshop on Structured Prediction for NLP*, pages 74–83, Online. 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. 2020a. 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 and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683, Online. Association for Computational Linguistics.

Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. GraphRel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, Florence, Italy. Association for Computational Linguistics.

Zhenhao He, Yuhong He, Qingyao Wu, and Jian Chen. 2020. Fg2seq: Effectively encoding knowledge for end-to-end task-oriented dialog. In *International Conference on Acoustics, Speech, and Signal Processing*.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *Meeting of the Association for Computational Linguistics*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

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*, pages 829–838, Online. 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*, pages 7999–8009, Online. 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)*, pages 1641–1651, Online. 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.

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.

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*, pages 3554–3559, Online. Association for Computational Linguistics.

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.

Trung Minh Nguyen and Thien Huu Nguyen. 2019. One for all: Neural joint modeling of entities and events. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6851–6858.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607.

Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. 2016. Bidirectional attention flow for machine comprehension. In *International Conference on Learning Representations*.

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 *Thirty-Second AAAI Conference on Artificial Intelligence*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks. In *International Conference on Learning Representations*.

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.

C Walker, S Strassel, J Medero, and K Maeda. 2006. Ace 2005 multilingual training corpus. *Progress of Theoretical Physics Supplement*, 110(110):261–276.

Kai Wang, Weizhou Shen, Yunyi Yang, Xiaojun Quan, and Rui Wang. 2020. Relational graph attention network for aspect-based sentiment analysis. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3229–3238, Online. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Zhiyuan Liu, Juanzi Li, Peng Li, Maosong Sun, Jie Zhou, and Xiang Ren. 2019. 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.

Xi Xiangyu, Wei Ye, Shikun Zhang, Quanxiu Wang, Huixing Jiang, and Wei Wu. 2021. Capturing event argument interaction via a bi-directional entity-level recurrent decoder. 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 210–219, 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.

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.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Tongtao Zhang, Heng Ji, and Avirup Sil. 2019. Joint entity and event extraction with generative adversarial imitation learning. *Data Intelligence*, 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.

Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. 2021. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14638–14646.