Event Extraction as Machine Reading Comprehension

Jian Liu[1,2,3], Yubo Chen[1,2], Kang Liu[1,2], Wei Bi[4], Xiaojiang Liu[4]
[1] National Laboratory of Pattern Recognition, Institute of Automation
Chinese Academy of Sciences, Beijing, 100190, China
[2] University of Chinese Academy of Sciences
[3] Beijing Jiaotong University, 100044, China
[4] Tencent AI Lab, Shenzhen, 518000, China
{jian.liu, yubo.chen, kliu}@nlpr.ia.ac.cn
{victoriabi, kieranliu}@tencent.com

Abstract

Event extraction (EE) is a crucial information extraction task that aims to extract event information in texts. Previous methods for EE typically model it as a classification task, which are data-hungry and suffer from the data scarcity problem. In this paper, we propose a new learning paradigm of EE, by explicitly casting it as a machine reading comprehension problem (MRC). Our approach includes an unsupervised question generation process, which can transfer event schema into a set of natural questions, followed by a BERT-based question-answering process to retrieve answers as EE results. This learning paradigm enables us to strengthen the reasoning process of EE, by introducing sophisticated models in MRC, and relieve the data scarcity problem, by introducing the large-scale datasets in MRC. The empirical results show that: i) our approach attains state-of-the-art performance by considerable margins over previous methods. ii) Our model is excelled in the data-scarce scenario, for example, obtaining 49.8% in F1 for event argument extraction with only 1% data, compared with 2.2% of the previous method. iii) Our model also fits with zero-shot scenarios, achieving 37.0% and 16% in F1 on two datasets without using any EE training data.

last_hidden_states

1 Introduction

Event extraction (EE), a crucial information extraction (IE) task, aims to extract event information in texts. For example, in a sentence S1 (shown in Figure 1 (a)), an EE system should recognize an `Attack` event[1], expressed by an *event trigger* *stabbed* with four *event arguments* — *Sunday* (Role=`Time`), *a protester* (Role=`Attacker`), *an officer* (Role=`Target`), and *a paper cutter* (Role=`Instrument`). EE is shown to benefit a wide range of applications including knowledge base augmentation (Ji and Grishman, 2011), document summarization, question answering (Berant et al., 2014), and others.

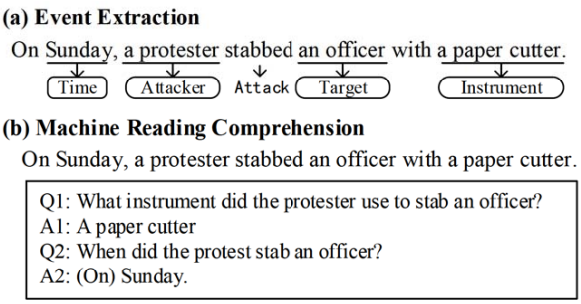

Figure 1: Comparison of the event extraction task and machine reading comprehension task.

In the current study, EE is mostly formulated as a *classification* problem, aiming to locate and categorize each event trigger/argument (Ahn, 2006; Li et al., 2013; Chen et al., 2015; Nguyen et al., 2016). Despite many advances, classification based methods are data-hungry, which require a great deal of training data to ensure good performance (Chen et al., 2017; Li et al., 2013; Liu et al., 2018a). Moreover, such methods generally cannot deal with *new* event types never encountered during training time (Huang et al., 2018).

In this particular study, we introduce a new learning paradigm for EE, shedding lights on tackling the above problems simultaneously. Our major motivation is that, essentially EE may be viewed as a machine reading comprehension (MRC) problem (Hermann et al., 2015; Chen et al., 2016) involving text understanding and matching, aiming to find event-specific information in texts. For example, in S1, the extraction of role-filler of `Instrument` is semantically equivalent to the following question-answering process (as shown in Figure 1 (b)):

Q1: *What* `Instrument` *did the protester use*

[1] According to the ACE event ontology.

Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, pages 1641–1651,
November 16–20, 2020.

to stab the officer? **A1:** *a paper cutter*. [2]
This implies new ways to tackle EE, which come with two major advantages: First, by framing EE as MRC, we can leverage the recent advances in MRC (e.g., BERT (Devlin et al., 2019)) to boost EE task, which may greatly strengthen the reasoning process in the model. Second, we may directly leverage the abundant MRC datasets to boost EE, which may relieve the data scarcity problem (This is referred to as *cross-domain data augmentation*). The second advantage also opens a door for zero-shot EE: for unseen event types, we can list questions defining their schema and use an MRC model to retrieve answers as EE results, instead of obtaining training data for them in advance.

To bridge MRC and EE, the key challenge lies in generating *relevant* questions describing an event scheme (e.g., generating Q1 for `Instrument`). Note we cannot adopt supervised question generation methods (Duan et al., 2017; Yuan et al., 2017; Elsahar et al., 2018), owing to the lack of aligned question-event pairs. Previous works connecting MRC and other tasks usually adopt human-designed templates (Levy et al., 2017; FitzGerald et al., 2018; Li et al., 2019b,a; Gao et al., 2019; Wu et al., 2019). For example, in QA-SRL (FitzGerald et al., 2018), the question for a predicate `publish` is always "Who published something?", regardless of the contexts. Such questions may not expressive enough to instruct an MRC model to find answers.

We overcome the above challenge by proposing an unsupervised question generation process, which can generate questions that are both relevant and context-dependent. Specifically, in our approach, we assume that each question can be decomposed as two parts, reflecting query topic and context-related information respectively. For example, Q1 can be decomposed as "*What instrument*" and "*did the protester use to stab the officer?*". To generate the query topic expression, we design a template-based generation method, combining role categorization and interrogative words realization. To generate the more challenging context-dependent expression, we formulate it as an *unsupervised* translation task (Lample et al., 2018b) (or style transfer (Prabhumoye et al., 2018)), which transforms a descriptive statement into a question-style expression, based on in-domain de-noising auto-encoding (Vincent et al., 2008) and cross-domain back-translation (Sennrich et al., 2016). Note the training process only needs large volume of descriptive statements and *unaligned* question-style statements. Finally, after the questions are generated, we build a BERT based MRC model (Devlin et al., 2019) to answer each of question and synthesize all of the answers as the result of EE.

每个问题包括两个部分：
主题，上下文相关的信息

To evaluate our approach, we have conducted extensive experiments on the benchmark EE datasets, and the experimental results have justified the effectiveness of our approach. Specifically, 1) in the standard evolution, our method attains state-of-the-art performance and outperforms previous EE methods by a margin (§ 4.2). 2) In the data-low scenario, our approach demonstrates promising results, for example, achieving 49.8% in F1 using 1% of training data, compared with only 2.2% in F1 of the previous EE method (§ 4.3). 3) Our approach also fits with zero-shot scenarios, achieving 37.0% and 16.6% in F1 on two datasets without using any EE training data (§ 4.4).

To sum up, we make the following contributions:

- We investigate a new formulation of EE, by framing it as an MRC problem explicitly. We show this new formulation can boost EE by leveraging both model and data in the area of MRC. Our work may encourage more works studying transfer learning from MRC to boost information extraction.
- We propose an unsupervised question generation method to bridge MRC and EE. Compared with previous works using templates to generate questions, our method can generate questions that are both topic-relevant and context-dependent, which can better instruct an MRC model for question-answering.
- We report on state-of-the-art performance on the benchmark EE dataset. Our method also demonstrate promising results in addressing data-low and zero-shot scenarios.

2 Related Work

Event Extraction. EE is a crucial IE task that aims to extract event information in texts, which has attracted extensive attention among researchers. Traditional EE methods employ manual-designed features, such as the syntactic feature (Ahn, 2006), document-level feature (Ji and Grishman, 2008), entity-level feature (Hong et al., 2011) and other features (Liao and Grishman, 2010; Li et al., 2013)

[2] Figure 1 (b) gives another example.

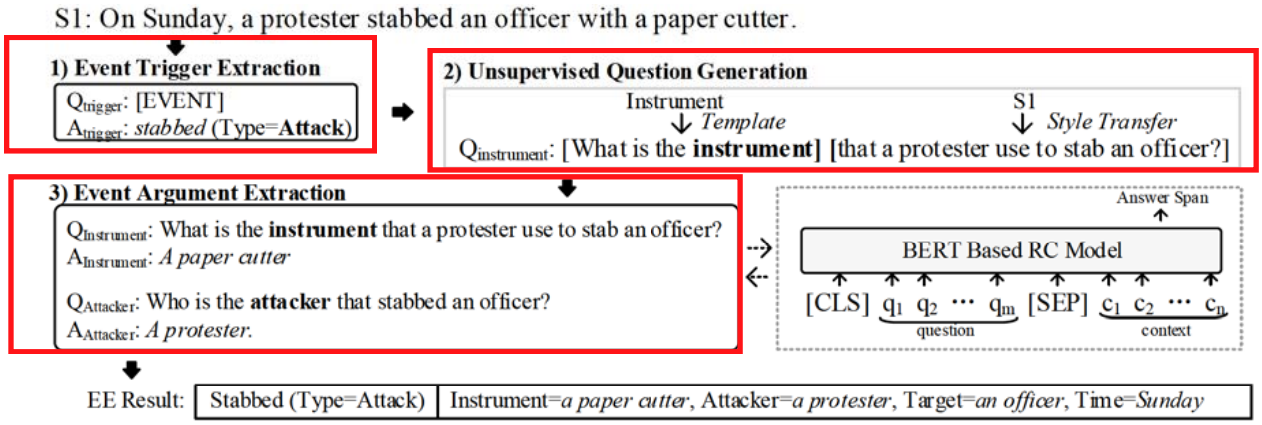

Figure 2: The overview of the proposed model RCEE. Given S1, RCEE first uses a special query [EVENT] to locate event trigger and predict the type. Then RCEE generates questions for each semantic role related to the predicted event type. Finally, RCEE answers each question and synthesizes all of the answers as the EE result.

for the task. Modern EE methods employ neural models, such as Convolutional Neural Networks (Chen et al., 2015), Recurrent Neural Networks (Nguyen et al., 2016; Sha et al., 2018), Graph Convolutional Neural Networks (Liu et al., 2018b, 2019b), and other advanced architectures (Yang and Mitchell, 2016; Liu et al., 2018a, 2019a; Nguyen and Nguyen, 2019; Zhang et al., 2019). Despite many advances, as mentioned in Introduction, most previous approaches formulate EE as a classification problem, which usually suffer from the data scarcity problem, and they generally cannot deal with new event types never seen at the training time.

MRC for Other Tasks. Our work also relates to works connecting MRC and other tasks, such as relation extraction (Levy et al., 2017; Li et al., 2019b), semantic role labeling (FitzGerald et al., 2018), named entity recognition (Li et al., 2019a), and others (Wu et al., 2019; Gao et al., 2019). Particularly, Du and Cardie (2020) adopt a similar idea to frames EE as MRC. But different from our work, most of the above methods (Levy et al., 2017; Li et al., 2019b; FitzGerald et al., 2018; Du and Cardie, 2020) adopt human-designed, context-independent questions, which may not provide enough contextual evidence for question-answering. Some works indeed do not adopt question-style queries (Li et al., 2019a; Gao et al., 2019). For example, Li et al. (2019a) use "Find organizations in the text" as a query command to find ORGANIZATION entity. The discrepancy between such non-natural "queries" and natural questions in MRC datasets may hinder effective transfer learning from MRC to the task. By contrast, our work aims to generate both relevant and context-related questions via an unsupervised question generation method.

3 The Approach

Our approach, denoted by RCEE (Reading Comprehension for Event Extraction), is visualized in Figure 2. Specifically, given a sentence S1, RCEE first identifies an event trigger "*stabbed*" and its event type `Attack`, on receiving a special query "[Event]". Secondly, RCEE generates a question for each semantic role corresponding to the event schema of `Attack`. Thirdly, RCEE builds an MRC model to answer each question as event argument extraction. Finally, RCEE synthesizes all of the answers as the final result of EE.

The technical details of RCEE are presented in the following. In the illustration, we denote a sentence as $c = \{c_1, \cdots, c_n\}$, and we structure the illustration as event trigger extraction, unsupervised question generation, event argument extraction, and the training procedure of RCEE.

3.1 Event Trigger Extraction

To extract event triggers, we use "[EVENT]" as a special query command, indicating finding all event triggers in texts[3]. The reason is that event triggers are usually verbs, and it is hard to design questions for them. Also note here this special query command enables event trigger and argument extraction share a same encoding model.

触发词和论元角色编码相同

[3]We have also tried questions like "*What events are mentioned in texts?*" and type-related questions like "*Which are ATTACK/DIE events?*" but found no improvement.

CATEGORY	ROLE	TEMPLS.
Time-related	`Time`	When
Place-related	`Place`	Where
Person-related	`Victim`, `Attacker`, ...	Who is the ROLE
General role	`Instrument`, `Target`, ...	What is the ROLE

Table 1: Role categorization and generation templates.

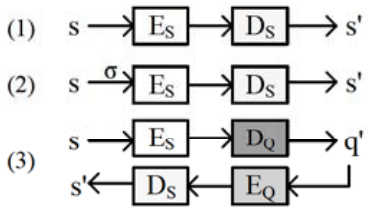

Figure 3: Illustration of (1) in-domain auto-encoding, (2) de-noising auto-encoding, and (3) cross-domain online-back translation. E_S (E_Q) and D_S (D_Q) are encoder and decoder in domain S (Q). σ denotes random noise such as word masking (Lample et al., 2018b).

Next, we adopt classification-based (instead of span-based method) for trigger extraction, considering that most triggers (over 95% in ACE) are single words, and span-based answer generation may be too heavy. Specifically, we first jointly encode "[EVENT]" with the sentence c to compute an encoded representation (we refer to § 3.3 for details). Then for each word c_i in c, we take its encoded representation as the input of a logistic regression model, and compute a vector o_{c_i} containing probabilities of different event types. Finally, the probability of the lth event type for c_i is $p(l|c_i) = o_{c_i}^{(l)}$, which is the lth element of o_{c_i}.

3.2 Unsupervised Question Generation

After trigger extraction, RCEE generates a set of questions according to the predicted event type. Here we assume each question can be composited as: 1) *query topic*, which reflects the relevance of a question, and 2) *question-style event statement*, which encodes the context-related information.

Question Topic Generation. We devise template-based methods for query topic generation. Note to make a question natural enough, we should consider different interrogative words for different semantic roles. For example, the query topic for the semantic role `Time` might be "When [...]", but for `Attacker` might be "Who [...]". With the above motivation, we first group semantic roles into different categories, and then design different templates for each category. Table 1 shows our categorization (i.e., time-related, place-related, person-related and general roles) and templates for the ACE 2005 event ontology. According to the table, the generated query topic for `Victim` is "*Who is the Victim*".

Question Contextualization. Question contextualization aims to generate the remaining question-style event statement. Here formulate it as an unsupervised translation task (Lample et al., 2018a,b), with a goal to maps *descriptive statement* (such as the sentence) to a *question-style statement*, with no parallel resources. It can also be viewed as style transfer (Prabhumoye et al., 2018). To achieve the goal, we first build large corpora of descriptive statements (denoted as S) and *unaligned* natural questions (denoted as Q)[4], and we restrict each instance in S a window of words centered at a verb, and each instance of Q a question removing interrogative words such as When/Where/Who/What. Second, following Lample et al. (2018b), we build two MT models: $P_{S\to Q}(q_s|s)$, which maps a descriptive statement $s \in S$ as a question-style statement q_s, and $P_{Q\to S}(s_q|q)$, which conducts the translation reversely. Each MT model includes an encoder and a decoder in the source and target domains respectively. For example, $P_{S\to Q}(q_s|s)$ has an encoder E_S in S, and a decoder D_Q in Q. Third, We train $P_{S\to Q}(q_s|s)$ and $P_{Q\to S}(s_q|q)$ jointly via in-domain auto-encoding, de-noising auto-encoding (Vincent et al., 2008), and cross-domain online-back translation (Sennrich et al., 2016), as shown in Figure 3. Finally, at the inference time, a window of words centered at the predicted trigger (denoted by s_x) is considered as input of $P_{S\to Q}(q_s|s)$, and we compute the question-style statement q_{s_x} via:

$$q_{s_x} = \arg\max_{q_{s_x}} P_{S\to Q}(q_{s_x}|s_x) \quad (1)$$

q_{s_x} is concatenated with the pre-generated query topic to generate the final question.

3.3 Event Argument Extraction

RCEE then performs event argument extraction as question answering, by using a BERT based MRC model. Let a question be $q = \{q_1, \cdots, q_m\}$

[4]In our approach, S contains sentences extracted from Wikipedia, and Q contains user-generated questions from a QA site `https://question.com/`. After filtering, S and Q have a size of 70M and 43M respectively

Learning Input Representations. We first encode q and c jointly to learn the input representations, by constructing an sequence "[CLS] q [SEP] c" as input of BERT. To further enhance the representation, we devise a new embedding, *word sharing embedding*, as the input of BERT, with a motivation that shared words of q and c are more likely to convey event information. Specifically, the word sharing embedding of a word w_i (in q or c) is:

$$p_{w_i} = \begin{cases} p_{sh} \in \mathbb{R}^{d_1} & \text{if } w_i \text{ is shared by } q \text{ and } c \\ p_{no} \in \mathbb{R}^{d_1} & \text{otherwise} \end{cases} \quad (2)$$

where p_{sh} and p_{no} are two embedding vectors getting updated during training. After encoding, we take the last hidden layer of BERT, $H_c^q \in \mathbb{R}^{N \times d_2}$, as the final representation of q and c, where $N = m + n + 2$[5], and d_2 designates BERT's hidden dimension.

Adaptive Argument Generation. Different from triggers, event arguments generation is tackled by span-based algorithms (Hermann et al., 2015), as they are usually entities and contain multiple words. While we note over 14% of semantic roles have *zero* or *multiple* arguments, we revise the existing algorithm to tackle the issue (shown in Algorithm 1). Specifically, given the joint representation H_c^q of q and c, we fist compute two probability vectors containing the start and end positions of the answer over every position in c:

$$p_{start} = \text{softmax}(H_c^q W_{start}) \quad (3)$$

$$p_{end} = \text{softmax}(H_c^q W_{end}) \quad (4)$$

where W_{start} and $W_{end} \in \mathbb{R}^{2d_4 \times 1}$ are model parameters. Then, we regard the special token "[SEP]" as "no-answer" indicator, and we only use start/end positions whose probabilities are higher than that of "[SEP]" to construct candidate answers. We adopt several heuristics regarding i) relative position of start/end index, length constraint, and likelihood threshold δ to filter out illegal answers. The new algorithm can generate both zero or more than one answers for a question. Additionally, when entity information is known (this setting is adopted in many approaches (Chen et al., 2015; Nguyen et al., 2016)), we further adopt *golden entity refinement*, which enforces answers have the same boundaries as ground-truth entities.

[5]For simplicity of illustration, we assume the output of BERT has a same length of "[CLS] q [SEP] c". In fact, BERT may split a word based on byte pair encoding.

通过设定阈值来抽取论元

Algorithm 1 Adaptive Argument Generation

```
procedure FUN(c, p_start, p_end)
    answer_list = []
    s_list ← filter_by_probability(p_start)
    e_list ← filter_by_probability(p_end)
    ▷ Construct candidate answers using s_list and e_list
    for each candidate (s_idx, e_idx) do
        ▷ s_idx should be ahead of e_idx
        ▷ length should less than 4
        if p_start[s_idx] + p_end[e_idx] > δ then
            ans = make_span(c, s_idx, e_idx)
            answer_list.add(ans)
        end if
    end for
    golden_entity_refinement(answer_list)
    return answer_list
end procedure
```

最后加上结果优化

3.4 Training

To train RCEE, we adopt a pre-training followed by fine-tuning strategy, which can jointly train a model using datasets of MRC and EE.

Pre-training Stage. In the pre-training stage, we train RCEE on MRC datasets, with a loss:

$$\mathcal{L}_{rc}(\theta) = \sum_{\langle c,q,a \rangle} \text{P}(a|c,q) \quad (5)$$

where $\langle c, q, a \rangle$ denotes an MRC example consisting of context c, query q, and answer a; $\text{P}(a|c,q)$ indicates the likelihood of the ground-truth answer a given c and q, which is defined as:

$$\text{P}(a|c,q) = \log p(g_s^a|c,q) + \log p(g_e^a|c,q) \quad (6)$$

where g_s^a and g_e^a are respectively the ground-truth start/end positions.

Fine-Tuning Stage. In the fine-tuning stage, we train RCEE on EE datasets with a loss:

$$\mathcal{L}_{ev}(\theta) = -\sum_e \left(\log p(g_e|w_e) + \sum_{r \in \mathcal{A}(g_e)} \text{P}(a_r|c_e, q_r) \right) \quad (7)$$

where e ranges over each event instance; w_e indicates the trigger of e; g_e indicates the event type of e; $Arg(e)$ designates the role set of g_e; r ranges over each rule. We adopt Adam (Kingma and Ba, 2014) to update parameters of RCEE.

Method	Trigger Ex.			Argument Ex.			Argument Ex.(O)		
	P	R	F1	P	R	F1	P	R	F1
JointBeam (Li et al., 2013)	73.7	62.3	67.5	64.7	44.4	52.7	-	-	-
DMCNN (Chen et al., 2015)	75.6	63.6	69.1	62.2	46.9	53.5	59.0†	54.8†	56.8†
JRNN (Nguyen et al., 2016)	66.0	73.0	69.3	54.2	56.7	55.4	57.5†	58.2†	57.9†
dbRNN (Sha et al., 2018)	74.1	69.8	71.9	66.2	52.8	58.7	58.4†	64.2†	61.2†
JMEE (Liu et al., 2018b)	**76.1**	71.3	73.7	66.8	54.9	60.3	59.8†	64.2†	62.0†
BERTEE	74.8†	73.9†	74.3†	70.5†	52.2†	60.6†	66.8†	62.6†	64.7†
RCEE_ER (ours)	75.6	**74.2**	**74.9***	63.0	**64.2**	**63.6***	**71.2**	**69.1**	**70.1***
RCEE_ER *w/o* DA (ours)	-	-	-	61.8	63.6	62.7	69.6	68.4	69.0

Table 2: Results of trigger extraction (Trigger Ex.), argument extraction (Argument Ex.), and argument extraction with golden triggers (Argument Ex.(O)). P, R and F1 stand for precision, recall, and f1-score respectively; † denotes our re-implementation; * denotes a significance level of $p = 0.05$.

4 Experiments

4.1 Experimental Setups

Datasets and Evaluation. Our experiments are conducted on the widely-used ACE 2005 benchmark[6], which defines 33 different event types and 35 semantic roles. We split the dataset as training, validating, and testing sets according to previous works (Li et al., 2013; Chen et al., 2015; Yang and Mitchell, 2016), and we also adopt precision (P), recall (R), and F1-score (F1) as evaluation metrics to ensure comparability. Significance tests are conducted using methods proposed by Yeh (2000) with a significance level of $p = 0.05$.

Implementation Details. We adopt BERT-Large, which has 24 layers, 1024 hidden units, and 16 attention heads, as our MRC model. Other hyper-parameters are tuned on the validating set via a grid search. Specifically, the dimension of word sharing embedding is set as 100 (from 10, 50, 100, 200, to 500). The answer prediction threshold δ is set as 0.3 (from [0.1, 0.2, .., 0.9]). The batch size is set as 10 (from 2, 5, 10, 15). The dropout rate is set as 0.5. We adopt SQuAD 2.0 (Rajpurkar et al., 2018) for cross-domain data argumentation (Our MRC model achieves 83.9% in F1). Implementations of unsupervised question generation are in supplement materials. Our code will be released at https://github.com/jianliu-ml/EEasMRC.

Baseline Models. We compare our model with: 1) JointBeam (Li et al., 2013), a state-of-the-art feature-based method for EE; 2) DMCNN (Chen et al., 2015) and 3) JRNN (Nguyen et al., 2016), two models adopting Convolution Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) respectively for EE; 4) dbRNN (Sha et al., 2018) and 5) JMEE (Liu et al., 2018b) two models exploring syntax information via RNNs and Graph Convolutional Neural Networks (GCNs) for EE. Joint EE models are also considered, including: 6) Joint3EE (Nguyen and Nguyen, 2019), which uses a unified architecture to predict entities and events; 7) JointTrans (Zhang et al., 2019), which adopts a left-to-right transaction-based method for EE. To further investigate whether the improvement are introduced by BERT representation, we also consider: 8) BERTEE, which adopts BERT representations but uses classification strategy for EE. Our model is denoted as RCEE and RCEE_ER ("ER" denotes with golden entity refinement). We use DA to indicate cross-domain data augmentation.

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

4.2 Standard Evaluation

In the standard evaluation, we consider two settings with 1) known entities, which is considered by many previous methods, and 2) unknown entities, which is a more realistic setting.

Results with Known Entities. Table 2 gives the results of trigger (Trigger Ex.) and argument extraction (Argument Ex.) with known entities. We also report on results of argument extraction with oracle triggers (Argument Ex.(O)), to exclude the potential error propagation from trigger extraction results. From the results, 1) RCEE_ER attains state-of-the-art performance, outperforming all baselines by considerable margins (+0.6% in trigger extraction; (+3.6% (5.4%)) in argument ex-

METHOD	G_E	P_E	ΔF1
JointBeam (2013)	52.7	41.8	↓10.9
DMCNN (2015)	56.8	48.0†	↓8.8
JMEE (2018b)	60.3	50.4†	↓9.9
BERTEE	60.6†	51.9†	↓8.7
Joint3EE (2019)	-	52.1	-
JointTrans (2019)	-	53.3	-
RCEE	63.6	**59.3***	↓4.3
RCEE *w/o* DA	62.7	58.7	↓4.0

Table 3: Results of argument extraction with unknown entities (P_E). ΔF1 indicates the performance gap compared with results with known entities (G_E).

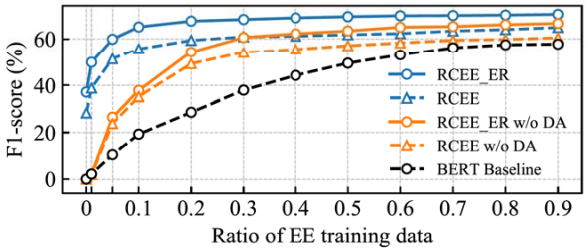

Figure 4: Results on different ratios of EE training data.

METHOD	1%	5%	10%	20%
DMCNN	-	8.7	16.6	23.7
dbRNN	-	8.1	17.2	24.1
BERTEE	2.20	10.5	19.3	28.6
RCEE	38.8	51.3	55.7	59.4
RCEE *w/o* DA	2.00	23.8	35.2	49.2
RCEE_ER	**49.8**	**59.9**	**65.1**	**67.6**
RCEE_ER *w/o* DA	2.20	26.5	37.8	54.1

Table 4: F1 score (%) on exploring the extremely data-scarce scenarios.

traction). 2) Especially, RCEE_ER outperforms BERTEE (which also use BERT representations) with over 5% in argument extraction, which indicates that the improvements are mainly from problem reformulation, rather than introducing BERT representations. 3) The high recall of RCEE_ER indicates that it can predict more examples than baselines, which may imply that RCEE_ER can tackle difficult cases that fail baseline models.

Results with Unknown Entities. Table 3 gives results with unknown entities. In this setting, classification-based methods need to identify entities first, thus we implement a BERT-base one for them[7]. Joint EE methods are also compared, which do not require entity information. We use RCEE for comparison, which excludes entity refinement. From the results, RCEE still demonstrates the best performance — it beats both classification based methods (over 9.3% in F1) and joint models (over 6.0%). By checking $\Delta F1$, we note RCEE relies relatively less on golden entities (-4.3% in F1 without them), but classification-based methods depend heavily on them, suffering from a drop of over 8% in F1 with the predicted entities.

4.3 Results in Data-Scarce Scenarios

Figure 4 compares models and BERTEE in data-scarce scenarios, and Table 4 gives results in the *extremely* data-low scenario ($\leq 20\%$ training data)[8]. From the results, our model demonstrates superior performance, for example, obtaining 49.8% in F1 with only 1% of EE training data, in comparison to 2.2% in F1 of BERTEE. We note the improvement comes from two aspects: 1) Data augmentation (DA). For example, DA improves +47.6% and +33.4% for RCEE_ER in experiments with 1% and %5 data according to Table 4. 2) Answer generation algorithm. Note RCEE_ER without DA still consistently outperforms BERTEE in data-low scenarios. This implies the answer generation algorithm is *data-efficient* than classification method. The reason might be that, the answer generation algorithm in our approach is position-based, which might be robust for unseen words. While the classification method in previous EE methods are largely word-based, which requires more labeled data.

4.4 Results in Zero-Shot Scenarios

Table 5 shows the results regarding zero-shot EE, where EE data is completely banned for training (Only using DA for model pre-training). To increase the persuasiveness of results, we adopt another dataset, FrameNet (Baker, 2014) (where frames are treated as meta event type) for evaluation. From the results: without any EE data, our model achieves 37% and 16.6% in F1 on ACE and FrameNet. This illustrates the effectiveness of our model handling unseen types.

[7]One tagger reaches 85.4%/85.9%/85.6% in P/R/F1, matching the state-of-the-art (Yang and Mitchell, 2016).

[8]To simplicity discussion, we assume golden triggers in the following experiments.

DATASET	MODEL	P	R	F1
ACE2005	RCEE	25.5	26.0	25.8
	RCEE_ER	38.2	35.8	37.0
FrameNet	RCEE	18.2	15.3	16.6

Table 5: F1 score (%) on exploring the zero-shot scenarios on ACE 2005 and FrameNet.

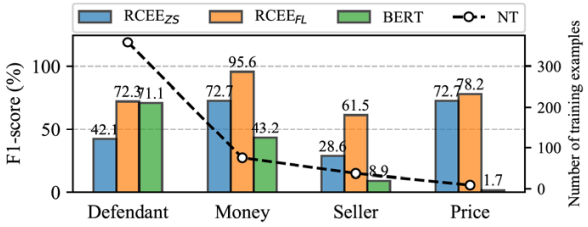

Figure 5: Performance on different roles. RCEE$_{ZS}$ and RCEE$_{FL}$ indicate zero-shot and full-training scenarios. NT denotes number of training data of a role.

(a) Putin were scheduled to **leave** ... *16 words omit* ... nations in Evian, France.
Role=`Destination` | G="Evian, France" | P=NONE

(b) Attempts by Laleh and Ladan to have ... *14 words omit* ... both of them could **die**.
Role=`Victim` | G="Laleh and Ladan" | P="them"

Table 6: Example error cases. Bold denotes trigger; G and P denote the ground-truth and predicted argument.

5 Further Discussion

5.1 Impact of Question Generation

We compare different question generation strategies: 1) QRole, which uses a role's name as query; 2) QCommand, which uses "Find the #Role" as query (Li et al., 2019a), and 3) QTemplate, which uses a template "What is the #ROLE in the #event_trigger event?" as query (FitzGerald et al., 2018). From the results, QRole, QCommand, and QTemplate achieve 60.1%, 64.9%, and 68.%5 in F1 in argument extraction; compared with 70.1% of our approach. We note the inferiority of those methods may lay in their poor expression ability. For example, in a sentence "The pair *flew* to Singapore last year after ...", QNAME uses "Time" as query; QCommand uses "Find the Time" as query; QTemplate uses "What is the Time in the flew event?" as query. While our approach directly generates a nearly perfect question "[When] do the pair fly to Singapore?" We provide more examples in supplement materials.

5.2 Performance on Different Roles

Figure 5 shows the performance of RCEE on different semantic roles, regarding four randomly selected roles with 1) plenty data, e.g. `Defendant` with 359 training examples; 2) medium-sized data, e.g. `Money` with 75 examples; and 3) limited data, e.g. `Seller` and `Price` with only 32 and 9 examples (rare roles). From the results, classification based methods, e.g. BERTEE, can achieve a good result for roles with plenty data, but their performance deteriorates seriously when a role has insufficient data. By comparison, our approach RCEE demonstrates excellent performance in handling rare roles, for example, obtaining 61.5% and 78.2% in F1 for `Seller` and `Price` (note `Price` has only 9 examples), in compared with 8.9% and 1.7% of BERTEE.

5.3 Error Analysis

We conduct error analysis in this section. One typical error is related to long-range dependency, accounting for 23.4% (here "long-range" denotes the distance between a trigger and an argument is $\geq$ 10). Table 6 (a) shows a case, where the argument *Evian, France* is about 20 words away from the trigger **leave**, making it difficult to identify the argument. 2) The second error relates to roles whose meaning are general, e.g., `Entity`, `Agent` — it is usually difficult to generate meaningful questions for these roles, causing 32.7% errors among all cases. 3) The third error relates to co-reference, which accounts for 17.2%. Considering the example in Table 6 (b), where **die** evokes a `Die` event with "Laleh" and "Ladan" fulfilling a semantic role `Victim`. Our model predicts "them" (two words ahead of **die**) as answer — though "them" is a reference of "Laleh and Ladan", it considered as an error according to current evaluations. This also raises the question of whether we should consider co-reference when we evaluate EE systems.

6 Conclusion and Future Work

In this paper, we take a fresh look at EE by casting it as an MRC problem. Our method includes an unsupervised question generation process which can generate both relevant and context-related questions, whose effectiveness is verified by empirical results. In the future, we would adapt our method to other IE tasks to study its application scope.

Acknowledgments

This work is supported by the Natural Key R&D Program of China (No.2018YFB1005100), the National Natural Science Foundation of China (No.U1936207, No.61976211, No.61806201) and the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006). This work is also supported by CCF-Tencent Open Research Fund, Beijing Academy of Artificial Intelligence (BAAI2019QN0301) and independent research project of National Laboratory of Pattern Recognition.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Collin Baker. 2014. FrameNet: A knowledge base for natural language processing. In *Proceedings of Frame Semantics in NLP: A Workshop in Honor of Chuck Fillmore (1929-2014)*, pages 1–5, Baltimore, MD, USA. Association for Computational Linguistics.

Jonathan Berant, Vivek Srikumar, Pei-Chun Chen, Abby Vander Linden, Brittany Harding, Brad Huang, Peter Clark, and Christopher D. Manning. 2014. Modeling biological processes for reading comprehension. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1499–1510, Doha, Qatar. Association for Computational Linguistics.

Danqi Chen, Jason Bolton, and Christopher D. Manning. 2016. A thorough examination of the CNN/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2358–2367, Berlin, Germany. Association for Computational Linguistics.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. 2017. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–419, Vancouver, Canada. 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. Event extraction by answering (almost) natural questions.

Nan Duan, Duyu Tang, Peng Chen, and Ming Zhou. 2017. Question generation for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 866–874, Copenhagen, Denmark. Association for Computational Linguistics.

Hady Elsahar, Christophe Gravier, and Frederique Laforest. 2018. Zero-shot question generation from knowledge graphs for unseen predicates and entity types. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 218–228, New Orleans, Louisiana. Association for Computational Linguistics.

Nicholas FitzGerald, Julian Michael, Luheng He, and Luke S. Zettlemoyer. 2018. Large-scale QA-SRL parsing. In *ACL*.

Shuyang Gao, Abhishek Sethi, Sanchit Agarwal, Tagyoung Chung, and Dilek Hakkani-Tur. 2019. Dialog state tracking: A neural reading comprehension approach. In *Proceedings of the 20th Annual SIGdial Meeting on Discourse and Dialogue*, pages 264–273, Stockholm, Sweden. Association for Computational Linguistics.

Karl Moritz Hermann, Tomás Kociský, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *NIPS*.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1148–1158, Portland, Oregon, USA. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018a. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations (ICLR)*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018b. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 333–342, Vancouver, Canada. 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 (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2019a. A unified mrc framework for named entity recognition.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019b. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. In *AAAI*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *AAAI Conference on Artificial Intelligence*.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019b. Neural cross-lingual event detection with minimal parallel resources. 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 738–748, Hong Kong, China. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. 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.

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 *AAAI Conference on Artificial Intelligence*.

Shrimai Prabhumoye, Yulia Tsvetkov, Ruslan Salakhutdinov, and Alan W Black. 2018. Style transfer through back-translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 866–876, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational 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 *AAAI Conference on Artificial Intelligence*.

Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. 2008. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 1096–1103, New York, NY, USA. ACM.

Wei Wu, Fei Wang, Arianna Yuan, Fei Wu, and Jiwei Li. 2019. Coreference resolution as query-based span prediction.

Bishan Yang and Tom M. Mitchell. 2016. Joint extraction of events and entities within a document context. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 289–299, San Diego, California. Association for Computational Linguistics.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *COLING 2000 Volume 2: The 18th International Conference on Computational Linguistics*.

Xingdi Yuan, Tong Wang, Caglar Gulcehre, Alessandro Sordoni, Philip Bachman, Saizheng Zhang, Sandeep Subramanian, and Adam Trischler. 2017. Machine comprehension by text-to-text neural question generation. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, pages 15–25, Vancouver, Canada. Association for Computational Linguistics.

Junchi Zhang, Yanxia Qin, Yue Zhang, Mengchi Liu, and Donghong Ji. 2019. 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-19*, pages 5422–5428. International Joint Conferences on Artificial Intelligence Organization.

A Implementation Details of Unsupervised Question Generation

Following Lample et al. (2018b), we use FastBPE to split each example into sub-word units, with a vocabulary size of $60k$. We implement both encoders and decoders as 4-layer transformers, where one layer is domain-specific for both the encoder and decoder and the rest are shared. Moreover, we use the standard hyper-parameter settings recommended by (Lample et al., 2018b). The input word embeddings are initialized as FastText vectors trained on the concatenation of the S and Q.

Negotiations between Washington and Pyongyang on Role = `Time/Place` (When/Where) did the negotiations between Washington and Pyongyang begin ?
founder Stelios Haji-Ioannou , who *set up* easyJet in 1995 and built Role = `Time/Place` (When/Where) did founder Stelios Ioannescu set up his company ?
divorce in September after their *marriage* broke down . Role = `Time/Place` (When/Where) did the divorce occur after their marriage ?
The total *purchase* cost is estimated at 300 Role = `Price` (What is the price) of the total cost of building a nuclear power plant ?
His wife will go on *trial* next week on charges of Role = `Defendant` (Who is the defendant) on trial next week?
Security Council for its 1990 *invasion* of Kuwait should be removed Role = `Attacker` (Who is the attacker) for its 1990 Gulf War ?
in U.S. troops for a *war* against Iraq even though it Role = `Attacker` (Who is the attacker) for a war against Iraq ?
Kuvaldin of a research center *funded* by former Soviet president Mikhail Role = `Organization` (What is the organization) of Kubidran University funded by ?

Table 7: Examples of generated questions. In each cell, the first line is the original sentence (event triggers are in italic); the second line is the semantic role; the third line is the generated question. () denotes the query topic generated by templates, and the remaining part is the query-style expression generated by our model.

During training, we reduce the coefficient of auto-encoding loss from 1.0 to 0.5 by 100K steps and to 0 by 300K steps. We cease training when the BLEU scores between back-translated and input questions stop improving, usually around 800K steps. For inference, we use a beam size of 5 and a language model to evaluate all the candidates to yield the best one.

B Generated Questions

Some generated questions are given in Table 7. Note these examples are directly taken from our model's output without any manual edition (We do not even add a question mark at the end of each question).