Exploring Entities in Event Detection as Question Answering

Emanuela Boros, Jose G. Moreno, Antoine Doucet

Exploring Entities in Event Detection as Question Answering*

Emanuela Boros[1][0000−0001−6299−9452] Jose G. Moreno[1,2][0000−0002−8852−5797]
Antoine Doucet[1][0000−0001−6160−3356]

[1] University of La Rochelle, L3i, F-17000, La Rochelle, France
{emanuela.boros,antoine.doucet}@univ-lr.fr
https://www.univ-larochelle.fr
[2] University of Toulouse, IRIT, UMR 5505 CNRS, F-31000, Toulouse, France
jose.moreno@irit.fr

Abstract. In this paper, we approach a recent and under-researched paradigm for the task of event detection (ED) by casting it as a question-answering (QA) problem with the possibility of multiple answers and the support of entities. The extraction of event triggers is, thus, transformed into the task of identifying answer spans from a context, while also focusing on the surrounding entities. The architecture is based on a pre-trained and fine-tuned language model, where the input context is augmented with entities marked at different levels, their positions, their types, and, finally, their argument roles. Experiments on the ACE 2005 corpus demonstrate that the proposed model properly leverages entity information in detecting events and that it is a viable solution for the ED task. Moreover, we demonstrate that our method with different entity markers is particularly able to extract unseen event types in few-shot learning settings.

Keywords: Event detection · Question answering · Few-shot learning.

1 Introduction

Event extraction (EE) is a crucial and challenging task of information extraction (IE) that aims at identifying the instances of specified types of events in a text, generally referred to as event detection (ED), and the detection and classification of the corresponding arguments (participants). For instance, according to the ACE 2005 annotation guidelines[3], an event is described as having the following characteristics:

– the *event mention* is an occurrence of an event with a particular type. This is usually a sentence or a phrase that describes an event; the *event trigger* is the word that most clearly expresses the event mention, e.g. *Attack*;

* This work has been supported by the European Union's Horizon 2020 research and innovation program under grants 770299 (NewsEye) and 825153 (Embeddia), and by the ANNA and Termitrad projects funded by the Nouvelle-Aquitaine Region.

[3] https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

– the *event argument* is an entity mention or temporal expression (e.g., *Crime*, *Job-Title*) that serves as a participant with a specific role in an event mention. Event arguments have an *entity type*, e.g. persons (PER), locations (LOC), organizations (ORG), etc.; and the *argument role* that is the relationship between an argument and the event in which it participates.

Following this description, from the sentence "***Police*** *have* *arrested* ***four people*** *in connection with the* *killings*.", an event extraction system should be able to recognize the word killings as a trigger for an event of type *Die*, with the person (PER) entity *Police* as an argument with the role of an Agent and the person *four people* as an argument of type Person, and the word arrested as a trigger for an *Arrest-Jail* event type with no arguments.

In this paper, we approach the task of event detection (ED) by studying the usage of entities in a recent and under-researched paradigm for the task of event detection (ED) by casting it as a question-answering (QA) problem.

There have been several deep learning-based major techniques applied for approaching the ED task while taking advantage of entity or argument information in the literature. First, systems extensively utilized linguistic analysis, entity information, entity coreference, and other knowledge resources to capture the discrete structures for ED, focusing on the combination of these discriminative features to build statistical models [7, 11, 13]. Next, neural-based approaches were based on convolutional and recurrent neural networks (CNNs and RNNs) that utilized effective feature representations from entity type embeddings [23, 4, 24, 22, 21, 25, 16].

Recent approaches adopt the usage of pre-trained language models [28]. Since BERT [5] broke records for several natural language processing (NLP) tasks (part-of-speech tagging, named entity recognition, etc.) and received a lot of attention, recent advances in ED imply architectures based on fine-tuning this type of models [31, 8, 29, 2], these methods holding the state of the art for ED.

Differently from these Transformer-based methods, where event and argument detection were considered as classification tasks, a new paradigm was introduced [6, 15] formulating EE as a question answering (QA)/machine reading comprehension (MRC[4]) task, where events can be extracted by responding to the 5W1H questions (who did what, when, where, why, and how). While these recent advances claim to cast the EE task as an MRC task [6, 15], they mostly focus on argument extraction as QA, while for ED, the models remain formulated as a sequential classification problem that aims at detecting event triggers of specific types.

Thus, in this paper, we focus on the event detection task, and we first cast it as a QA task with the possibility of multiple answers, in the case where more than one event is present in the text. By approaching it as a QA model, not only are we able to leverage the recent advances in MRC, we also avoid the classification based-methods that can either require lots of training data and are challenged by the annotation cost or data scarcity.

[4] In one view, the recent tasks titled MRC can also be seen as the extended tasks of question answering (QA).

Second, we take advantage of the presence of entities for extracting the events[5], considering that informative features can be brought by additional entity markers for better distinguishing the event triggers. We agree that "Entities of the consistent type normally participate in similar events as the same role." [7].

In addition, modeling the task as QA can improve the ED task in regard to this challenge due to the fact that the answers are only considered in relation to the context and the question, which could reduce trigger ambiguity. Furthermore, compared to classification based-methods [6, 15, 4, 15, 12] that generally lack this ability, we demonstrate that our proposed QA models are more effective in few-shot scenarios by showing that they are able to extract unseen event types. The work of [9] is distinguished in the literature, where the authors prove that zero-shot learning for detecting events can be possible and efficient. They proposed to leverage existing human-constructed event schemas and manual annotations for a small set of seen types, and transfer the knowledge from the existing event types to the extraction of unseen event types. We consider this paper as our reference method for the few-shot learning setting, and we prove that modeling the ED task as QA with entity information can obtain higher performance results.

Our proposed method with entity information obtains state-of-the-art results when compared with previous models that utilize entity or argument information. Moreover, these methods could foster further research and help to study transfer learning from QA models to boost the performance of existing information extraction systems. Furthermore, compared to classification based-methods that lack this ability, we demonstrate that our proposed QA models are more effective in few-shot scenarios by showing that they are able to extract unseen event types.

Next, we continue with the related work in Section 2, and we detail the QA model with entity markers in Section 3. The experimental setup and the results are presented in Section 4. We provide a discussion of the results by analyzing the output in Section 5 and we draw conclusions in Section 6.

2 Related Work

Event Detection with Entity Information In the context of event detection, some works made use of gold-standard entities in different manners. Higher results can be obtained with gold-standard entity types [23], by concatenating randomly initialized embeddings for the entity types. A graph neural network (GNN) based on dependency trees [25] has also been proposed to perform event

[5] We note here that event extraction generally depends on previous phases as, for example, named entity recognition, entity mention coreference, and classification. Thereinto, the named entity recognition is another hard task in the ACE evaluation and not the focus of this paper. Therefore, we will temporarily skip the phase and instead directly use the entities provided by ACE, following previous work [7, 14, 10, 4, 15, 12].

detection with a pooling method that relies on entity mentions aggregation. Arguments provided significant clues to this task in the supervised attention mechanism proposed to exploit argument information explicitly for ED proposed by [17]. Other methods that took advantage of argument information were joint-based approaches.

The architecture adopted by [18] was jointly extracting multiple event triggers and event arguments by introducing syntactic shortcut arcs derived from the dependency parsing trees. [7]'s cross-entity feature-based method extracted events by using gold standard cross-entity inference in order to take advantage of the consistency of entity mentions while achieving efficient sentence-level trigger and argument (role) classification. [13] utilized the contextual entities in a joint framework based on a structured prediction that extracted triggers and arguments together so that the local predictions can be mutually improved.

Approaches presented by [23] and [4] experimented with the integration of entities in ED models based on CNNs. These models utilized effective feature representations from pre-trained word embeddings, position embeddings as well as entity type embeddings. [24] improve the previous model proposed by [23] by taking into account the possibility to have non-consecutive n-grams as basic features instead of continuous n-grams.

A different technique was explored by [1] and it consisted in marking the entities in the relation extraction task and by studying the ability of the Transformer-based neural networks to encode relations between entity pairs. They identified a method of representation based on marking the present entities that outperform previous work in supervised relation extraction. [20] also explored the use of pre-trained neural models into the relation validation problem by explicitly using a triplet-sentence representation with marked entities, proving that the relation extraction performance could be further improved by using this additional information. Furthermore, [2] also proposed the use of pre-trained neural models in a BERT-based classification-based architecture for detecting events.

Event Detection as Question Answering While QA for event detection is roughly under-researched, Transformer-based models have led to striking gains in performance on MRC tasks recently, as measured on the SQuAD v1.1[6] [27] and SQuAD v2.0[7] [26] leaderboards.

A recent work proposed by [6] introduced this new paradigm for event extraction by formulating it as a QA task, which extracts the event triggers and arguments in an end-to-end manner. For detecting the event, they considered an approach based on BERT that is usually applied to sequential data. The task of ED is a classification-based method where the authors designed simple fixed templates as in *what is the trigger*, *trigger*, *action*, *verb*, without specifying the event type. For example, if they chose *verb* template, the input sequence would

[6] SQuAD v1.1 consists of reference passages from Wikipedia with answers and questions constructed by annotators after viewing the passage

[7] SQuADv2.0 augmented the SQuAD v1.1 collection with additional questions that did not have answers in the referenced passage.

be: [CLS] *verb* [SEP] sentence [SEP]. Next, they use a sequential fine-tuned BERT for detecting event trigger candidates.

Another recent paper [15] also approaches the event extraction task as a question answering task, similar to the [6] method. The task remains classification-based (instead of the span-based QA method) for trigger extraction, jointly encode [EVENT] with the sentence to compute an encoded representation, as in the approach proposed by [6] where the special token was *verb* or *trigger*.

3 Event Question Answering Model with Entity Positions, Types, and Argument Roles

We formulate the ED task as a QA task, where, for every sentence, we ask if a particular event type is present, and we expect a response with an event trigger, multiple event triggers, or none. Our model extends the BERT [5] pre-trained model which is a stack of Transformer layers [28] that takes as input a sequence of subtokens, obtained by the WordPiece tokenization [30] and produces a sequence of context-based embeddings of these subtokens.

To feed a QA task into BERT, we pack both the question and the reference text into the input, as illustrated in Figure 1. The input embeddings are the sum of the token embeddings and the segment embeddings. The input is processed in the following manner: token embeddings (a [CLS] token is added to the input word tokens at the beginning of the question and a [SEP] token is inserted at the end of both the question and the reference text) and segment embeddings (a marker indicating the question or the reference text is added to each token). This allows the model to distinguish between the question and the text.

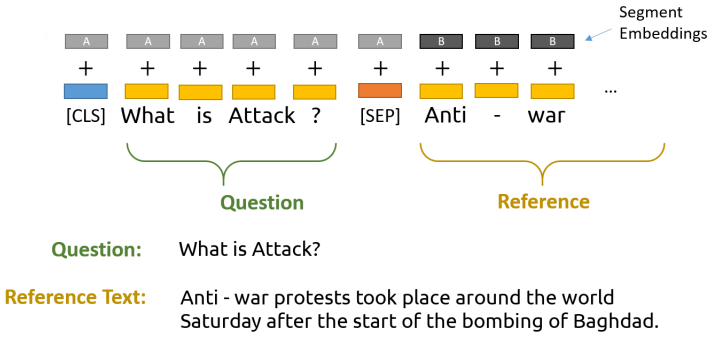

Fig. 1. Example of input modification to fit the QA paradigm for a sentence that contains an event of type *Attack*. The question is separated by [SEP] token from the reference text that contains the event trigger war.

To fine-tune BERT for a QA system, a start vector and an end vector are introduced. A linear layer is added at the top of BERT layers with two outputs

for the start and end vectors of the answer. The probability of each word being the start or end word is calculated by taking a dot product between the final embedding of the word and the start or end vector, followed by a Softmax over all the words. The word with the highest probability value is considered. This method differs from the event detection approaches presented by [6] and [15] where the models are classification-based, instead of the span-based QA.

Next, for every type of event **[Event Type]** (*Demonstrate, Die, Attack*, etc.), we formulate the question by automatically generating them using the following template:

What is the [Event Type] ?

An example for a sentence containing an *Attack* event is illustrated in Figure 1. We also consider questions that do not have an answer in the case where an event of a specific type is not present in the sentence. When there is more than one event of the same type in a sentence, we consider that the question has multiple answers. From the n best-predicted answers, we consider all those that obtained a probability higher than a selected threshold (established on the development set). When the predicted chunks are self-contained, we consider only the first predicted event trigger. For example, if the noun chunks *assault* and *air assault* are predicted, only *assault* is considered.

Next, for adding entity information, we augment the input data with a series of special tokens. Thus, if we consider a sentence $x = [x_0, x_1, \ldots, x_n]$ with n tokens, we augment x with two reserved word pieces to mark the beginning and the end of each event entity or argument mention in the sentence.

Next, we propose three types of markers: (1) *Entity Position Markers*, e.g. <E> and </E> where E represents an entity of any type, (2) *Entity Type Markers*, e.g. <PER> and </PER> where PER represents an entity of type Person, and (3) if the event argument roles are known beforehand, *Argument Role Markers*, e.g. <Agent>, </Agent> where Agent is an event argument role. Thus, we modify the following sentence:

"***Police** have arrested **four people** in connection with the killings.*"

where killings is a trigger for a *Die* event, and arrested is a trigger for an *Arrest-Jail* event, *Police* is one of the participants, a person (*PER*) with the argument role of an *Agent*, and *four people* is also a person entity (*PER*) with the *Person* argument role. The modified sentences with the three types of markers are:

(1) "*<E> **Police** </E> have arrested <E> **four people** </E> in connection with the killings.*"

(2) "*<PER> **Police** </PER> have arrested </PER> **four people** </PER> in connection with the killings.*"

(3) "*<Agent> **Police** </Agent> have arrested <Person> **four people** </Person> in connection with the killings.*"

Further, an ED system should detect in the presented sentence, the trigger word killings for an event of type Die (this event has two arguments *Police* and *four people*) and arrested for an event of type Arrest-Jail (this event has no

arguments). For the *Argument Role Markers*, if an entity has different roles in different events that are present in the same sentence, we mark the entity with all the argument roles that it has.

4 Experiments

Table 1. Evaluation of our models and comparison with state-of-the-art systems for event detection on the blind test data. The models with ♣ utilized gold standard entity mentions. The models with ♡ utilized gold standard arguments. Statistical significance is measured with McNemar's test. * denotes a significant improvement at $p \leq 0.01$.

Approaches	**P**	**R**	**F1**
MaxEnt with local features♣ [12]	74.5	59.1	65.9
Cross-entity♣ [7]	72.9	64.3	68.3
DMCNN♣ [4]	75.6	63.6	69.1
Word CNN♣ [23]	71.8	66.4	69.0
Joint RNN♣ [21]	66.0	73.0	69.3
BERT-QA-base-uncased	68.4	70.5	69.5
BERT-base [6]	67.1	73.2	70.0
Non-Consecutive CNN♣ [22]	–	–	71.3
Attention-based♣♡ [16]	78.0	66.3	71.7
BERT_QA_Trigger [6]	71.1	73.7	72.3
Graph CNN♣ [25]	77.9	68.8	73.1
BERT-QA-base-uncased + Entity Position Markers♣	78.0	70.7	74.2*
RCEE_ER♣ [15]	75.6	74.2	74.9
BERT-QA-base-uncased + Entity Type Markers♣	78.5	77.2	77.8*
BERT-QA-base-uncased + Argument Role Markers♡	83.2	80.5	81.8*

The evaluation is conducted on the ACE 2005 corpus provided by ACE program[8]. For comparison purposes, we use the same test set with 40 news articles (672 sentences), the same development set with 30 other documents (863 sentences) and the same training set with the remaining 529 documents (14,849 sentences) as in previous studies of this dataset [10, 14]. The ACE 2005 corpus has 8 types of events, with 33 subtypes (e.g. the event type *Conflict* has two subtypes *Attack, Demonstrate*). In this paper, we refer only to the subtypes of the events, without diminishing the meaning of main event types.

Evaluation Metrics Following the same line of previous works, we consider that a trigger is correct if its event type, subtype, and offsets match those of a reference trigger. We use Precision (P), Recall (R), and F-measure (F1) to evaluate the overall performance.

[8] `https://catalog.ldc.upenn.edu/LDC2006T06`

Hyperparameters We used the Stanford CoreNLP toolkit[9] to pre-process the data, including tokenization and sentence splitting[10]. For fine-tuning the BERT-based models, we followed the selection of hyperparameters presented by [5]. We found that 3×10^{-5} learning rate and a mini-batch of dimension 12 for the *base* models provided stable and consistent convergence across all experiments as evaluated on the development set. The maximum sequence length is set to 384 and the document stride of 128. For selecting the event triggers, we generate $n = 10$ candidates, and we use the same threshold for all the experiments, with a value of 0.2 that was decided on the development set.

General Evaluation In Table 1, we present the comparison between our model and state-of-the-art approaches that utilised entity or argument information.

We compare with the MaxEnt-based model with local features in [12], the cross-entity feature-based method extracted events by using gold standard cross-entity inference [7] and the models proposed by [4], [23], [22], and the joint framework with bidirectional RNNs [21] that experimented with the integration of entities in ED models based on a CNN-based architectures.

We also compare with the method proposed by [16] that also exploited entity information explicitly for ED via supervised attention mechanisms, and the graph CNN by [25] that investigated a CNN based on dependency trees for ED with pooling method that relied on entity mentions to aggregate the convolution vectors.

We also compare with the models where the task has been approached as a QA task but still formulated as a sequential classification problem that aims at locating trigger candidates, the fine-tuned baseline BERT-base-uncased and the BERT_QA_Trigger [6], and the RCEE_ER (Reading Comprehension for Event Extraction, with *ER* that denotes that the model has golden entity refinement) [15].

When compared with the previous state-of-the-art models that included entity information, except for the RCEE_ER method, our models that use either the positions or the types of the entities bring a considerable improvement in the performance of trigger detection. It is clear that further marking the entities with their types can increase both precision and recall, balancing the final scores.

It is noteworthy that, while entities can be present in the entire document, arguments can only surround event triggers. Knowing the argument roles beforehand brings further improvements, we assume that an important reason for this is that, since the arguments are present only around event triggers, this could help the language model to be more aware of the existence of an event or multiple events in a sentence.

[9] `http://stanfordnlp.github.io/CoreNLP/`

[10] The code is available at `https://github.com/nlpcl-lab/ace2005-preprocessing` as it consists of the same pre-processing as utilized in several other papers [23, 21].

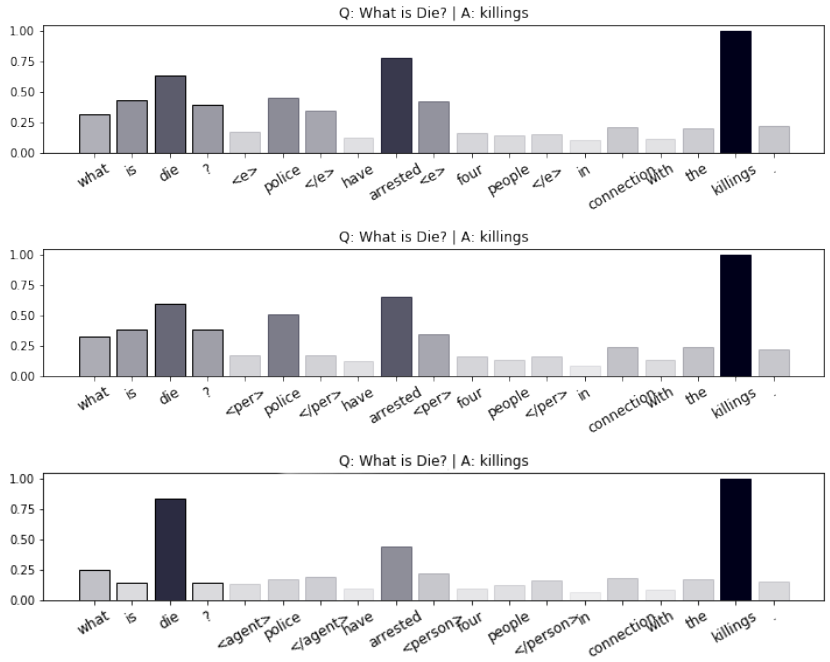

Fig. 2. An example for the *Die* event triggered by killings with three types of markers: *Entity Position, Entity Type*, and *Argument Role Markers*.

5 Discussion

5.1 Trigger Ambiguity Analysis

For a deeper analysis of the impact of entity information, we leverage the gradients in our proposed models to efficiently infer the relationship between the question, context, and the output response. [3] studied the identifiability of attention weights and token embeddings in Transformer-based models. They show that the self-attention distributions are not directly interpretable and suggest that simple gradient explanations are stable and faithful to the model and data generating process.

Thus, as applied by [19], to get a better idea of how well each model memorizes and uses memory for contextual understanding, we analyze the connectivity between the desired output and the input. This is calculated as:

$$\text{connectivity}(t, \tilde{t}) = \left|\left| \frac{\partial y_k^{\tilde{t}}}{\partial x^t} \right|\right|_2$$

where t is the time index, $\tilde{t}$ the output time index, and the result is the magnitude of the gradient between the logits for the desired output $y_k^{\tilde{t}}$ and the input x^t. The

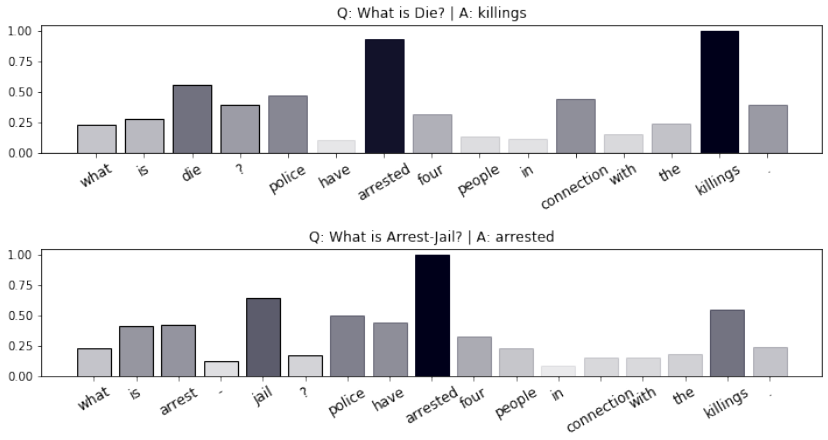

Fig. 3. An example of a sentence that contains two events: *Die* event triggered by the word killings and *Arrest-Jail* event triggered by arrested. The model used is BERT-QA-base-uncased.

Fig. 4. [CLS] representation of each sentence in the test set that contains at least an event for BERT-QA-base-uncased, BERT-QA-base-uncased + *Entity Position Markers+*, BERT-QA-base-uncased + *Entity Type Markers+*, and BERT-QA-base-uncased + *Argument Role Markers*.

connectivity is computed with respect to both start position and end position of the answer, then it is normalized, and it is visible as saliency maps for every word in Figures 3 and 2[11].

By looking at the gradients in Figure 3, where two events of different types are present, we can observe, in the upper part of the figure, that while the model sees the word killings and arrested as impactful, it also sees the words *police, connection* as impactful and selects an answer in that neighborhood. Even though both trigger candidates killings and arrested have a clear impact due to their gradient values, by looking at the probability values, killings is recognized with a 99.4% probability, while arrested obtained a probability of 2.3×10^{-7}, value that is lower than our selected threshold 0.2. In the lower part

[11] The sentence is lowercased for the *uncased* models.

of the figure, for the question *What is Arrest-Jail?*, the words *die, police,* killings clearly influence the choice of the answer arrested.

In Figure 2, we present the same sentence with the three types of input modifications: *Entity Position Markers, Entity Type Markers*, and *Argument Role Markers*, with the *What is Die?* question and the correct answer killings. In the upper part of the figure, where the sentence has been augmented with the entity position markers <E> and </E>, we notice that the words that impact the most in the result are killings along with *die*, arrested, and *police*. In this case, one can also see that the end marker </E> contributed too.

In the middle part of the figure, where the sentence has been augmented with the entity position markers <PER> and </PER> for the two entities *police* and *four people*, the influence of other words as in *die*, arrested, and *police* slightly decreased. In the bottom part of the image, the gradients of these words are visibly reduced.

When the sentence is augmented with argument roles, <Agent>, </Agent>, <Person> and </Person>, the noise around the correct answer has noticeably diminished, being reduced by the additional markers. The most impactful remaining words are the word *die* in the question and the correct answer killings.

In order to analyze the quality of the sentence representations, we extract the [CLS] representation of each sentence for BERT-QA-base-uncased and for BERT-QA-base-uncased + *Argument Role Markers*. Then, we plot these representations in two spaces where the labels (colors of the dots) are the event types, as illustrated in Figure 4. On the right-hand side of the figure, where argument role markers are used, it is clear that the sentence representations clusters are more cohesive than when no entity information is considered (left-hand side), thus confirming our assumption regarding the importance of the entity informative features in a QA system.

5.2 Evaluation on Unseen Event Types

In the first scenario, we follow the same strategy as [6] where we keep 80% of event types (27) in the training set and 20% (6) unseen event types in the test set. More exactly, the unseen event types were chosen randomly, and they are: *Marry, Trial-Hearing, Arrest-Jail, Acquit, Attack*, and *Declare-Bankruptcy*. Table 2 presents the performance scores of our models for the unseen event types.

Table 2. Evaluation of our models on unseen event types. The models with ♣ utilized gold standard entity mentions. The models with ♡ utilized gold standard arguments.

Approaches	**P**	**R**	**F1**
BERT-QA-base-uncased (*not trained on ACE 2005*)	0.7	8.3	1.3
BERT-QA-base-uncased	47.7	26.7	31.1
BERT-QA-base-uncased + *Entity Position Markers*♣	44.0	47.5	37.3
BERT-QA-base-uncased + *Entity Type Markers*♣	53.6	54.4	50.4
BERT-QA-base-uncased + *Argument Role Markers*♡	83.3	47.4	53.6

We compare with BERT-QA-base-uncased which is our baseline that selects an event trigger in a sentence without being trained on ACE 2005 data. Since the majority of the models in Table 1 are classification-based in a sequential manner, they are not capable of handling unseen event types, and thus, we were not able to obtain performance values. From the results, without any event annotation, the BERT-QA-base-uncased obtains a low F1 value (1.38%). We observe that the performance values increase proportionally to the specificity of the markers. Thus, it is not surprising that the highest values are obtained when the argument roles are marked, also obtaining the highest precision.

In a second scenario, we consider larger amounts of unseen events, and we follow the strategy proposed by [9], where out of the total number of event types (33), we select the top-N most popular event types as seen, while the rest remain unseen. N is set as 1, 3, 5, 10 respectively. We perform experiments in four settings (A, B, C, and D). Table 3 shows the types that were selected for training in each experiment setting.

Table 3. Seen types in each experiment setting as proposed by [9].

Setting	N	Seen Event Types
A	1	Attack
B	3	Attack, Transport, Die
C	5	Attack, Transport, Die, Meet, Arrest-Jail
D	10	Attack, Transport, Die, Meet, Sentence, Arrest-Jail, Transfer-Money, Elect, Transfer-Ownership, End-Position

Table 4. Evaluation of our models on unseen event types (Hit@1 as in [9]). The models with ♣ utilized gold standard entity mentions. The models with ♡ utilized gold standard arguments.

Approaches	**Settings**			
	A	B	C	D
Huang et al. [9]	3.9	7.0	20.0	33.4
BERT-QA-base-uncased + *Entity Positions Markers*♣	2.3	4.9	18.8	21.7
BERT-QA-base-uncased + *Entity Type Markers*♣	2.3	8.8	21.8	25.8
BERT-QA-base-uncased + *Argument Role Markers*♡	2.4	10.0	26.2	32.0

Table 4 presents the performance scores of our models for the unseen event types. We focus on showing the effectiveness of our methods juxtaposed with the results of [9]. We first observe, for each model, that the performance values improve as the number of seen events types. Second, one can notice that the scores also increase proportionally to the specificity of the markers.

6 Conclusions and Perspectives

In this paper, we utilized a recent and under-researched paradigm for detecting events by modeling the ED as a QA task with the addition of entity and argument information. The questions were simplified to a pre-defined list with a question for every type of event present in the dataset, which allows the model to predict multiple events in a sentence. The additional informative features brought by the presence of entities and the argument roles in the same context of the events considerably increased the performance of the model, achieving state-of-the-art results. Moreover, this type of model that utilizes the entity information leveraged the ambiguity of the event triggers and demonstrate potential in detecting unseen event types.

In future work, we will focus on approaching the entity and argument detection tasks, in order to analyze the influence of the predicted event arguments and the error propagation from this task to the downstream event detection task. Furthermore, we will consider approaching both event extraction sub-tasks (ED and argument detection and classification) in a joint QA-based architecture for alleviating the aforementioned issue concerning the diffusion of detection errors.

References

1. Baldini Soares, L., FitzGerald, N., Ling, J., Kwiatkowski, T.: Matching the blanks: Distributional similarity for relation learning. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2895–2905. Association for Computational Linguistics, Florence, Italy (Jul 2019). https://doi.org/10.18653/v1/P19-1279, https://www.aclweb.org/anthology/P19-1279
2. Boros, E., Moreno, J.G., Doucet, A.: Event detection with entity markers. In: Hiemstra, D., Moens, M., Mothe, J., Perego, R., Potthast, M., Sebastiani, F. (eds.) Advances in Information Retrieval - 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28 - April 1, 2021, Proceedings, Part II. Lecture Notes in Computer Science, vol. 12657, pp. 233–240. Springer (2021). https://doi.org/10.1007/978-3-030-72240-1_20, https://doi.org/10.1007/978-3-030-72240-1_20
3. Brunner, G., Liu, Y., Pascual, D., Richter, O., Ciaramita, M., Wattenhofer, R.: On identifiability in transformers. In: International Conference on Learning Representations (2019)
4. Chen, Y., Xu, L., Liu, K., Zeng, D., Zhao, J.: 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). pp. 167–176 (2015)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
6. Du, X., Cardie, C.: Event extraction by answering (almost) natural questions. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language

Processing (EMNLP). pp. 671–683. Association for Computational Linguistics, Online (Nov 2020), https://www.aclweb.org/anthology/2020.emnlp-main.49
7. Hong, Y., Zhang, J., Ma, B., Yao, J., Zhou, G., Zhu, Q.: Using cross-entity inference to improve event extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. pp. 1127–1136. Association for Computational Linguistics (2011)
8. Hong, Y., Zhou, W., Zhang, J., Zhou, G., Zhu, Q.: Self-regulation: Employing a generative adversarial network to improve event detection. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 515–526 (2018)
9. Huang, L., Ji, H., Cho, K., Dagan, I., Riedel, S., Voss, C.: Zero-shot transfer learning for event extraction. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 2160–2170. Association for Computational Linguistics, Melbourne, Australia (Jul 2018). https://doi.org/10.18653/v1/P18-1201, https://www.aclweb.org/anthology/P18-1201
10. Ji, H., Grishman, R., et al.: Refining event extraction through cross-document inference. In: ACL. pp. 254–262 (2008)
11. Li, P., Zhu, Q., Zhou, G.: Argument inference from relevant event mentions in chinese argument extraction. In: ACL (1). pp. 1477–1487 (2013)
12. Li, Q., Ji, H., Huang, L.: Joint event extraction via structured prediction with global features. In: ACL (1). pp. 73–82 (2013)
13. Li, W., Cheng, D., He, L., Wang, Y., Jin, X.: Joint event extraction based on hierarchical event schemas from framenet. IEEE Access **7**, 25001–25015 (2019)
14. Liao, S., Grishman, R.: Using document level cross-event inference to improve event extraction. In: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics. pp. 789–797. Association for Computational Linguistics (2010)
15. Liu, J., Chen, Y., Liu, K., Bi, W., Liu, X.: Event extraction as machine reading comprehension. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 1641–1651 (2020)
16. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 1789–1798 (2017)
17. Liu, S., Chen, Y., Liu, K., Zhao, J.: Exploiting argument information to improve event detection via supervised attention mechanisms. In: 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017). pp. 1789–1798. Vancouver, Canada (2017)
18. Liu, X., Luo, Z., Huang, H.: Jointly multiple events extraction via attention-based graph information aggregation. arXiv preprint arXiv:1809.09078 (2018)
19. Madsen, A.: Visualizing memorization in rnns. Distill (2019). https://doi.org/10.23915/distill.00016, https://distill.pub/2019/memorization-in-rnns
20. Moreno, J.G., Doucet, A., Grau, B.: Relation classification via relation validation. In: Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6). pp. 20–27 (2021)
21. Nguyen, T.H., Cho, K., Grishman, R.: Joint event extraction via recurrent neural networks. In: Proceedings of NAACL-HLT. pp. 300–309 (2016)
22. Nguyen, T.H., Fu, L., Cho, K., Grishman, R.: A two-stage approach for extending event detection to new types via neural networks. ACL 2016 p. 158 (2016)

23. Nguyen, T.H., Grishman, R.: Event detection and domain adaptation with convolutional neural networks. In: ACL (2). pp. 365–371 (2015)
24. Nguyen, T.H., Grishman, R.: Modeling skip-grams for event detection with convolutional neural networks. In: Proceedings of EMNLP (2016)
25. Nguyen, T.H., Grishman, R.: Graph convolutional networks with argument-aware pooling for event detection. In: Thirty-Second AAAI Conference on Artificial Intelligence (AAAI 2018) (2018)
26. Rajpurkar, P., Jia, R., Liang, P.: Know what you don't know: Unanswerable questions for squad. arXiv preprint arXiv:1806.03822 (2018)
27. Rajpurkar, P., Zhang, J., Lopyrev, K., Liang, P.: Squad: 100,000+ questions for machine comprehension of text. arXiv preprint arXiv:1606.05250 (2016)
28. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
29. Wang, X., Han, X., Liu, Z., Sun, M., Li, P.: Adversarial training for weakly supervised event detection. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 998–1008 (2019)
30. Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., et al.: Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)
31. Zhang, T., Ji, H., Sil, A.: Joint entity and event extraction with generative adversarial imitation learning. Data Intelligence **1**(2), 99–120 (2019)