Towards Focused and Connected Document-Level Event Extraction

Anonymous ACL submission

Abstract

Document-level event extraction (DEE) is indispensable when events are naturally described in the form of a document. Although previous methods have made great success on DEE, they are limited by two bottlenecks: losing focus and losing the connection. In this paper, to break through the above bottlenecks, we annotated a new dataset, named WIKIEVENT++, towards focused and connected DEE. Besides, we propose two different models to approach this task: the extractive model and the generative model. Experimental results verify the effectiveness of our proposed methods. We further present a promising case study to explore the performance bottleneck for this task. Data and code will be released at `http://anonymized` to advance the research on document-level event extraction.

1 Introduction

Event extraction (EE) aims to identify instantiated events, which include triggers with pre-defined types and their corresponding arguments, from narrative texts (Grishman et al., 2005). Previous studies (Chen et al., 2015; Nguyen et al., 2016; Yang and Mitchell, 2016; Chen et al., 2017; Huang et al., 2018; Yang et al., 2019; Liu et al., 2020) focused on the sentence-level EE. Benefiting from introducing neural network models and pre-trained language models for EE, these studies have achieved great success.

However, an event often goes beyond the sentence boundaries. As a result, extracting events from a single sentence will cause incomplete and uninformative event information (Li et al., 2021). For example, as shown in Figure 1, the "`Attacker`" role of the "`Conflict.Attack.DetonateExplode`" event is "`the Taliban`" in S_1, while its trigger is "`explosion`" in S_3. In such a case, cross-sentence argument extraction is needed. Otherwise, some extracted arguments with the pronoun form (e.g. "`they`" in S_4) will result in uninformative extraction. To solve such problem, Du and Cardie (2020a) and Du et al. (2020) focused on the document-level event role filling based on the MUC-4 dataset (Grishman and Sundheim, 1996). Ebner et al. (2020); Zhang et al. (2020) and Wei et al. (2021) made efforts on the implicit crossing-sentence arguments linking task based on the RAMS dataset (Ebner et al., 2020). Li et al. (2021) proposed a conditional generation model for document-level event argument extraction and achieve star-of-the-art results on the WIKIEVENTS (Li et al., 2021) dataset.

Although these studies have made great contributions to document-level event extraction (DEE), current methods still have two limitations: **losing the focus** and **losing the connection**. In detail, there usually are core events with other peripheral events in a document (Choubey et al., 2018; Hamborg et al., 2019). Compared with peripheral events, core events can provide key information of the document (Liu et al., 2018). As the example shown in Figure 1, the core events are "`Conflict.Attack.DetonateExplode`" triggered by "`explosion`" and "`Life.Die`" triggered by "`killed`". And there is a peripheral event "`Conflict.Attack.Unspecified`" triggered by "`shot`". However, current event extraction methods treat core events and peripheral events equally and fail to figure out the core events. As a result, the major event information in a document will be missed or polluted. We call this problem as **losing the focus**. Secondly, some events described in a document can refer to the same real-world event. As shown in the running example, "`Life.Die`" events triggered by "`killed`" in the S_1 and S_2 are coreferential events. Meanwhile, their arguments "`More than 100 members`" and "`126 people`"

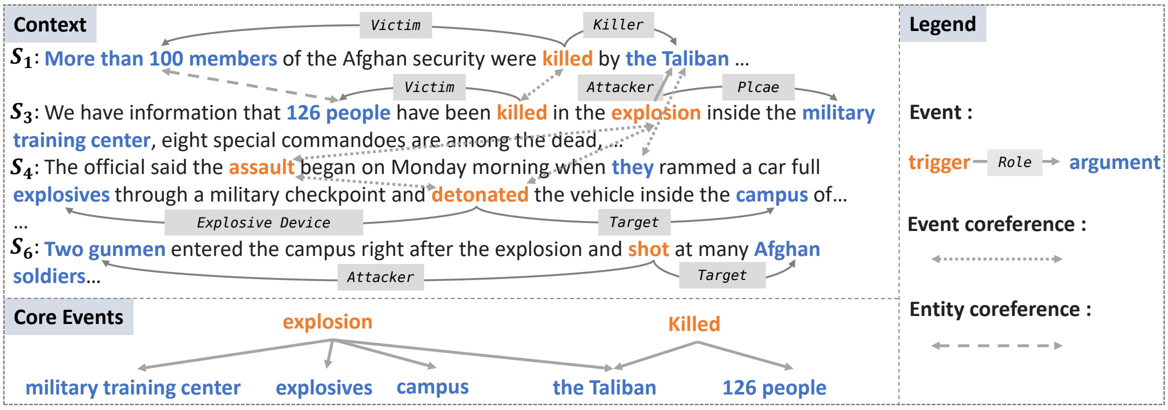
Context
Victim
Killer
S1: More than 100 members of the Afghan security were killed by the Taliban ...
Victim
Attacker
Plcae
S3: We have information that 126 people have been killed in the explosion inside the military training center, eight special commandoes are among the dead, ...
S4: The official said the assault began on Monday morning when they rammed a car full explosives through a military checkpoint and detonated the vehicle inside the campus of...
...
Explosive Device
Target
S6: Two gunmen entered the campus right after the explosion and shot at many Afghan soldiers...
Attacker
Target
Core Events
explosion
Killed
military training center
explosives
campus
the Taliban
126 people
Legend
Event :
trigger
Role
argument
Event coreference :
Entity coreference :

Figure 1: The given article mainly describes a "Conflict.Attack.DetonateExplode" event triggered by "explosion" and a "Life.Die" event triggered by "Killed". Words in blue represent arguments and words in red are triggers. The solid line denotes an entity plays in a role in an event and the dotted line indicates event/entity coreference relationships.

are coreferential entities. However, current event extraction methods extracted such connected events separately and fail to merge the coreferential events and coreferential entities, which is called **losing the connection**.

In this paper, we make the following efforts on the aforementioned issues and try to achieve focused and connected document-level event extraction. First, we construct a new document-level event extraction dataset, named WIKIEVENTS++, since the current existing datasets do not support this DEE task. Specifically, we annotate all occurrences of event coreference and core event annotation upon the WIKIEVENTS dataset (Li et al., 2021). Totally, we annotate 2,861 event clusters and 372 core events from 3,951 instantiated events in 246 documents. Besides, to accommodate the event extraction evaluation at the document level, we introduce new evaluation metrics, which consider the event clusters and entity clusters.

Second, to extract the core events in a document and build the connection between events, we approach the DEE task in two different manners: extractive model and generative model. In detail, the extractive model consists of a series of span extraction modules (entity extraction and event detection), pairwise classification modules (entity coreference, event coreference and event role identification) and core event detection. To obtain richer representations, we train the extractive model in a multi-task learning manner. Furthermore, we explore a generative model, based on the seq2seq framework (Sutskever et al., 2014). Compared with the extractive model, the proposed generative model does not require multiple pipe-lined operations for DEE, like entity/trigger extraction and entity/event coreference resolution, etc. As a result, the error propagation in the extractive model would be alleviated. Experimental results on the proposed dataset, WIKIEVENTS++, verify the effectiveness of our proposed methods.

The major contributions of this paper can be summarized as follows:

- We investigate the focused and connected document-level event extraction, which is unexplored before. For this task, we build a new DEE dataset, named WIKIEVENTS++.

- We propose two different ways: extractive model and generative model on this challenging task. The experimental results verify the effectiveness of the proposed two models on document-level reasoning and also explore the performance bottleneck of this task.

2 Related Work

2.1 Sentence-level Event Extraction

In recent years, most studies in event extraction focus on the sentence-level and achieves great success based on deep learning solutions (Chen et al., 2015; Nguyen et al., 2016; Yang et al., 2019; Chan et al., 2019; Yang et al., 2019; Liu et al., 2020). These studies are mainly based on the benchmark dataset, ACE 2005 (Doddington et al., 2004), a large-scale dataset with complete event annotation. In the ACE formulation, event extraction consists

Dataset	#Doc	#EventType	#ArgType	DocLevel	Trigger	EntityCoref	EventCoref	CoreEvent
ACE 2005 (Doddington et al., 2004)	599	33	35	✗	✓	✓	✓	✗
KBP 2017 (Getman et al., 2017)	167	18	20	✗	✓	✗	✓	✗
MUC-4 (Grishman and Sundheim, 1996)	1700	5	5	✓	✗	✗	✗	✗
ChiFinAnn (Zheng et al., 2019)	32040	5	35	✓	✗	✗	✗	✗
RAMS (Ebner et al., 2020)	9124	139	65	✓	✓	✗	✗	✗
WIKIEVENTS (Li et al., 2021)	246	50	59	✓	✓	✓	✗	✗
WIKIEVENTS++ (Ours)	246	50	59	✓	✓	✓	✓	✓

Table 1: A comparison between WIKIEVENTS++ and other existing widely-used EE datasets. "#Doc" indicates the number of annotated documents, "#EventType" denotes the number of event types, and "#RoleType" represents the number of event role types. Meanwhile, "DocLevel" denotes the event is described in a document-level or not, "Trigger" indicates including trigger annotation or not, "EntityCoref" denotes including entity coreference annotation or not, "EventCoref" represents including event coreference annotation or not, and "CoreEvent" denotes including core events annotation or not.

of two main subtasks: event detection (identify triggers with specific event types) and event argument extraction (identify the arguments the role types).

2.2 Document-level Event Extraction

As real-world events are often described across multiple sentences in a document, DEE is essential for event semantic understanding. The earliest DEE work can be traced back to the release of the MUC-4 datasets(muc, 1992), in which a document-level event role filler extraction task is defined. Recent studies explore this task by manually designing linguistic features (Patwardhan and Riloff, 2009; Huang and Riloff, 2011, 2012) or neural contextual representation (Chen et al., 2020; Du et al., 2020; Du and Cardie, 2020a). To investigate the arguments-scattering and multi-events in DEE, Zheng et al. (2019) release a large-scale document-level event extraction dataset, named Chinese financial announcements (ChFinAnn), and model DEE as an event table filling task. Following this setting, Zheng et al. (2019) propose Doc2EDAG, a directed acyclic graphs generation with entity-based path expanding. Xu et al. (2021) propose GIT, a graph neural network for entity encoding and a global memory mechanism for event decoding. Yang et al. (2021b) propose DE-PPN, a multi-granularity non-autoregressive decoder for multi-events generation. Although these methods achieved great success, they are still limited to the DEE in specific fields and a no-trigger formulation.

To explore the general field of document-level EE, Ebner et al. (2020) published the RAMS dataset, which annotated the triggers and its corresponding cross-sentences arguments within a five-sentence window. A two-step approach (Zhang et al., 2020) is proposed for argument linking by detecting implicit argument across sentences. Yang et al. (2021a) propose an event-aware hierarchical encoder for multi-sentence argument linking. Li et al. (2021) extend this task and compile a new benchmark dataset WIKIEVENTS which annotate cross-sentences arguments with informative mentions from Wikipedia articles. Then, Li et al. (2021) propose a conditional generation method for document-level informative arguments extraction. Although these studies make great success on DEE, they have two major limitations: losing focus and losing the connections. In this paper, we investigate the focused and connected document-level event extraction.

3 Datasets

3.1 Dataset Construction

To achieve focused and connected DEE, we annotate a new dataset, named WIKIEVENTS++. Specifically, we aim to build such a dataset not only containing event mention annotation (trigger and arguments), but also including core events and event coreference annotations. Since many current existing EE datasets have provided event mention annotations, we choose WIKIEVENTS (Li et al., 2021) as our base dataset and further annotated event coreference and core events. Annotators are asked to annotate coreferential event mentions to form event clusters and then identify the core events from these clusters. Note that the core events are usually mentioned many times in the document. Each document is annotated by two annotators independently. Once the annotation results are inconsistent, a third one will be involved for final annotation to ensure the consistency of annotation results. We used the BRAT (Stenetorp et al., 2012) interface for online annotation.

	#Doc	#Event	#EventCluster	#CoreEvent
Train	206	15.73	11.40	1.54
Dev	20	17.25	11.80	1.20
Test	20	18.25	13.75	1.45

Table 2: Statistics for the WIKIEVENTS++ dataset. "#Doc" denotes the number of documents. "#Event" denotes the average number of events in a document. "#EventCluster" denotes the average number of event clusters in a document. "#CoreEvent" denotes the average number of core events in a document.

3.2 Dataset Comparison

We compare WIKIEVENTS++ with several widely used event extraction dataset in Table 1. ACE 2005 (Doddington et al., 2004) is the most widely used sentence-level EE dataset with complete event annotation. KBP 2017 (Getman et al., 2017) is a sentence-level EE dataset released by Text Analysis Conference. MUC-4 (Grishman and Sundheim, 1996) is constructed with a fixed set of event types and associated five role types. ChFinAnn (Zheng et al., 2019) is a large-scale document-level event extraction dataset based on the Chinese financial announcements with five financial event types. The Roles Across Multiple Sentences (RAMS) (Ebner et al., 2020) make argument annotation in a five-sentence window around trigger words. WIKIEVENTS (Li et al., 2021) annotate cross-sentences arguments with informative mentions from Wikipedia articles. From Table 1, we can observe that the proposed dataset, WIKIEVENTS++, includes the most complete annotation for exploring the DEE task.

3.3 Dataset Stastics

The detailed statistics of the WIKIEVENTS++ dataset are presented in Table 6. We can observe that documents in the WIKIEVENTS dataset usually contain multiple granular events. These instantiated events form multiple event chains and revolve around a few core events.

4 Methodology

We formulate the task of focused and connected DEE in two different manners: extractive model and generative model. The extractive model consists of a series of modules, which are organized in a multi-task learning framework. The generative model frames this task as core events generation under an encoder-decoder learning paradigm.

In this section, we first present the formalization of the proposed DEE task. Then, we introduce the proposed extractive model for this DEE task. Finally, we describe the proposed end-to-end generative model.

4.1 Task definition

Given an input document comprised of N_c tokens $\mathcal{D} = \{c_i\}_{i=1}^{N_c}$, where N_c is the number of tokens in the document, the DEE task aims to extract core events where each event contains arguments with specific role types. For the extractive paradigm, there are some important subtasks typically including: **Entity Extraction**, which seeks to identify entities with pre-defined entity types from the document $\mathcal{D}$; **Event Detection**, which is a task to identify event triggers with pre-defined event types from the document $\mathcal{D}$; **Event Argument Extraction**, which aims to identify the arguments of an event and classify the roles that those arguments play; **Entity Coreference**, which is a task to resolve all mentions in the document $\mathcal{D}$ that refer to the same real-world entity; **Event Coreference**, whose goal is to determine which event mentions in the document $\mathcal{D}$ refer to the same real-world event; and **Core Event Detection**, which is a task to find events that are most relevant to the main content of the document $\mathcal{D}$.

4.2 Extractive Model

Figure 2 illustrates the workflow of the proposed extractive model for focused and connected DEE, which consists of three key modules: span extraction, pairwise classification and core event detection.

4.2.1 Encoding

Given a document $\mathcal{D} = \{c_i\}_{i=1}^{N_c}$ with N_c tokens, these tokens are first projected to the continuous vector space by using the pretrained word embedding. Then, word embeddings of these tokens, $[\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_{N_c}]$, are fed into an encoder to obtain the contextualized representations. In this paper, we adopt the Transformer (Vaswani et al., 2017) as the primary context encoder.

4.2.2 Span Extraction

Following Shi and Lin (2019), we model the entity extraction and the event detection as typical sequence tagging tasks, which identify the starting and ending position of each trigger or entity with their specific types. Through span extraction,

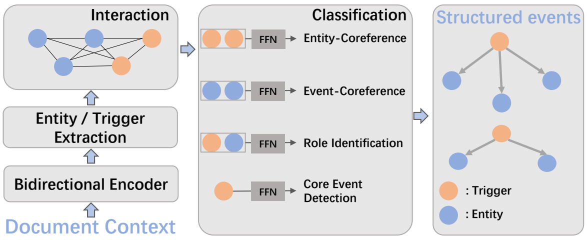

Figure 2: The workflow of the proposed extractive model for DEE.

we can obtain extracted triggers $T = \{t_i\}_{i=1}^{N_t}$ and entities $E = \{e_i\}_{e=1}^{N_e}$.

4.2.3 Global-Aware Interaction

To dynamically capture the interaction among all extracted spans (triggers and entities), following Zheng et al. (2019), we employ a Transformer model as the global-aware encoder. Specifically, given an extracted entity e_i with its span covering j-th to k-th tokens, we conduct a max-pooling operation over these token-level embedding to get the local embedding $\mathbf{h}_i^e \in \mathbb{R}^d$. Similar operation is also conduct on the triggers and we can get the i-th trigger representation $\mathbf{h}_i^t \in \mathbb{R}^d$. Then, we assemble entity type information and event type information with these extracted entities and triggers, respectively, and these assembled representations are fed into the global-aware encoder to facilitate the interaction between them. Note that, to inform the sentence order, we add the extracted entity and trigger representations with sentence-level position embeddings before feeding them into the global-aware encoder.

4.2.4 Pairwise Classification

There are three different relationships among these extracted entities and triggers: entity coreference (entity-entity), event coreference (trigger-trigger) and role identification (trigger-entity). To identify these relationships, we model these candidate pairs in a unified framework. For the role identification, given the global-aware i-th trigger representation $\mathbf{h}_i^t$ and j-th entity representation $\mathbf{h}_j^e$, we follow Yu et al. (2020) and build the pairwise representation as:

$$R_{i,j} = [\mathbf{h}_i^t; \mathbf{h}_j^e; \mathbf{h}_i^t \odot \mathbf{h}_j^e] \tag{1}$$

where $\odot$ denotes element-wise multiplication. Then, the pairwise representation $R(i, j)$ is fed into a feed-forward networks (FFN) for event role identification. Concretely, the predicted role type can be obtained by:

$$\mathbf{p}_{i,j}^{role} = \mathrm{softmax}(R_{i,j}\mathbf{W}_{role}) \tag{2}$$

where $\mathbf{W}_{role} \in \mathbb{R}^{d \times N_{role}+1}$ is learnable parameters, and N_{role} is the number of predefined roles.

Similarly, given the entity-entity pairs or trigger-trigger pairs, the entity coreference or event coreference prediction can be obtained by:

$$\mathbf{p}_{i,j}^{coref} = \mathrm{softmax}(R_{i,j}\mathbf{W}_{coref}) \tag{3}$$

where $\mathbf{W}_{coref} \in \mathbb{R}^{d \times 2}$.

4.2.5 Core Event Detection

To build the focused DEE system, core event detection is essential. For each extracted event, we use an FFN as the score function to detect core event, which can be denoted as:

$$\mathbf{p}_i^{core} = \mathrm{softmax}(\mathbf{h}_i^t\mathbf{W}_{core}) \tag{4}$$

where $\mathbf{W}_{core} \in \mathbb{R}^{d \times 2}$ is learnable parameters.

4.2.6 Multi-Task Training

We train the extractive model is in a manner of multi-task learning. We hypothesize that joint learning these tasks can result in richer representations and better performance.

$$L = l_{sp} + l_{coref} + l_{role} + l_{core} \tag{5}$$

where l_{sp}, l_{coref}, l_{role} and l_{core} denotes the loss of span extraction, coreference relationship classification, role identification and core event detection, respectively.

During training, we utilize both ground-truth entities and triggers for pairwise classification. While at inference, our model identifies entity and trigger firstly and then classifies the relationship for each pair. This gap between training and inference will cause error-propagation problems. To mitigate such a problem, we leverage the scheduled sampling (Bengio et al., 2015) for training.

4.3 Generative Model

We introduce an end-to-end generative model by transferring the extraction of core events into a sequence prediction, which is shown in Figure 3. Our generative model is based on an encoder-decoder pre-trained language model, BART (Lewis et al., 2020), which can generate a sequence given an input context. Specifically, a Transformer-based directional encoder is used to learn the feature for the

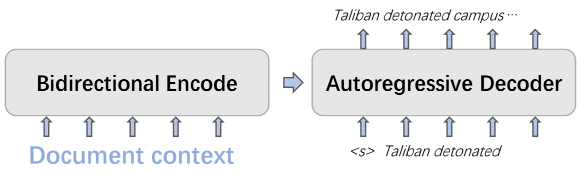

Figure 3: The overview of the generative model for DEE.

input $\mathcal{D}$, a Transformer-based left-to-right decoder is used for generating tokens. Specifically, take the “`Life.Die`” type event as example, given the input document $\mathcal{D}$, the expected output is based on the following templete: “ `<Role:Victim> died at <Role:Place> place killed by <Role:Killer>`” where “`<Role>`” is the placeholder filled by ground truth arguments. In the case where there are multiple core events in a document, we connect the sequence with a semicolon. During training, the generative model is trained by minimizing the negative loglikelihood of the generated sequence and ground truth sequence. During inference, we can get the sequential event by the generative process and finally obtain the structured events by post-processing.

5 Experiments and Analysis

In this section, we carry out experiments with the aim of answering the following research questions:

1. How well do our proposed models perform, in comparison with the baselines?

2. How does each module perform and each design work in the extractive model?

3. What is the performance bottleneck of the proposed extractive model and generative model.

In the remainder of this section, we describe baselines, evaluation metrics and experimental settings.

5.1 Baselines and Model Variations

For extracting the core events with informative arguments from a document, we adopt the baseline models as follows: **Seq** (Shi and Lin, 2019), which introduces a BERT-based BIO-styled sequence labeling model for argument identification. **QA** (Du and Cardie, 2020b), which is a QA-based model for document-level event argument extraction. To investigate the impact of input sentence length on performance, we adopt sentence-level encoder (short for “Sent”) and document-level encoder (short for “Doc”), respectively, for these baseline models and our extractive model.

5.2 Evaluation Metrics

Supported by different datasets, there are different evaluation criteria for the task of DEE. In this work, we define the task of DEE as extracting core events with connection in a document. We evaluate the document-level core events extraction in two metrics: coreferential mention F1 (Coref F1) and informative mention F1 (Infor F1). For the Coref F1, we consider an argument span in an extracted core event to be correct if the extracted argument is coreferential with the gold-standard argument as used in (Ji and Grishman, 2008). For the Infor F1, we consider an argument span to be correct if the extracted argument is the most informative mention in the entire document (Li et al., 2021). To consider the connection (entity coreference and event coreference) in the extractive model, we follow (Huang and Peng, 2021) and introduce two metrics: DocTri and DocArg. DocTri is used to evaluate the event clusters which contain coreference events with trigger span and event types. DocArg is used to evaluate the argument clusters which contain arguments with spans, role types and entity coreference. Details of the evaluation metric are presented in the Appendix.

5.3 Implementation Details

For the extractive model and baselines, we adopt roberta-large (Liu et al., 2019), a transformer-based pretrained language model, as the encoder. For the sentence-level encoder, we set the maximum length of sentences as 128. For the document-level encoder, we set the maximum length of the input context to 512 while the sliding window is used for splitting the document if the context length exceeds 512. For the generative model, we adopt BART-large (Lewis et al., 2020) as the encoder-decoder language model for generation. During training, we employ the AdamW optimizer (Kingma and Ba, 2014) with the learning rate 2e-5 for training 50 epochs and pick the best parameters by the validation score on the development set.

5.4 Main Results

We test our model on the test set of WIKIEVENTS++, the golden informative arguments are denoted as the target prediction for the **Seq** and **QA** baselines. Table 3 shows

Models	Coref			Infor		
	Prec.	Rec.	F1	Prec.	Rec.	F1
Seq-Sent	52.06	24.40	33.22	44.55	11.84	18.70
Seq-Doc	56.31	28.02	37.42	43.40	14.42	21.65
QA-Sent	38.44	26.00	31.02	51.81	10.39	17.30
QA-Doc	**57.07**	26.46	36.16	51.85	13.27	21.13
Extractive-Sent (Ours)	49.78	27.29	35.26	30.93	16.85	21.82
Extractive-Doc (Ours)	46.77	**32.58**	38.41	**54.81**	17.87	26.96
BART-Gen (Ours)	56.64	30.92	**40.00**	35.71	**22.73**	**27.78**

Table 3: Overall precision (P), recall (R) and F1 scores (F1) evaluated under document-level metrics (Coref F1 and Infor F1) for core events extraction on the WIKIEVENTS++ test set.

Models	Entity-C			Tri-C			DocTri			DocArg		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
Extractive-Sent	82.34	80.46	81.39	60.20	45.66	51.93	59.49	32.55	42.08	38.44	26.00	31.02
Extractive-Doc	84.62	80.86	82.70	64.74	46.42	54.07	60.48	38.11	46.76	33.87	33.60	33.74

Table 4: Results of each module in the extractive model on the WIKIEVENTS++ test set.

the comparison between our model and baseline methods under the Coref F1 and Infor F1 evaluation metrics. From the results, we can observe that: (1) Extracting informative arguments of core events from a document is extremely challenging as the extraction performance of all models drops significantly. We suspect that the inferior performance is due to the following reasons: Firstly, handling the long context is extremely challenge [1] which asks for the model's ability to capture the long-distance dependency among spans in a context. Secondly, extracting core events with their arguments is extremely challenge[2] which needs document-level relational reasoning among a multitude of candidate events and entities. (2) The generative model achieves the best performance on both two evaluation metrics. The performance indicates that the encoder-decoder generative framework for DEE is more effective. (3) For the extractive model and baselines, the model based on the document-level encoder performs better than that based on the sentence-level encoder, which indicates the importance of document-level modeling for the DEE task.

5.5 Performance of Each Module

For exploring the performance bottleneck for the extractive model, we test the performance for each module in the extractive model. The results are shown in Table 4. Note that Entity-C, Tri-C means the classification evaluation for entity extraction and event detection, respectively. DocTri and DocArg are the document-level metrics, which can evaluate our extractive DEE model on event coreference clusters and argument coreference clusters, respectively. From the results, we can observe that the F1-score of entity extraction and event detection on the WIKIEVENTS++ dataset achieve an acceptable performance. Note that the best F1 score for the entity extraction and event detection under the ACE 2005 datasets are around 90.3 and 75.2, respectively (Lin et al., 2020). We suspect that this gap is due to the scale of training data. Besides, we find that the inferior performance under the DocTri and DocArg evaluation and we conjecture that entity coreference and event coreference are extremely challenging tasks. Therefore, modeling entity-entity, event-event and event-entity pairwise dependencies may be the main bottleneck of the proposed extractive model.

5.6 Ablation Studies

In this section, to verify the effectiveness of each design of our proposed extractive model, we conduct ablation studies that are evaluated on the test set of the WIKIEVENTS++ dataset: (1) ***-MultiLearn***, which means that replace the multi-task learning with a pipline-based formulation. (2) ***-GlobalInter*** indicates removing the Transformer-based global interaction layer. (3) ***-SchSamp***,

[1] Average 793 tokens per document.

[2] Average 17 events per document.

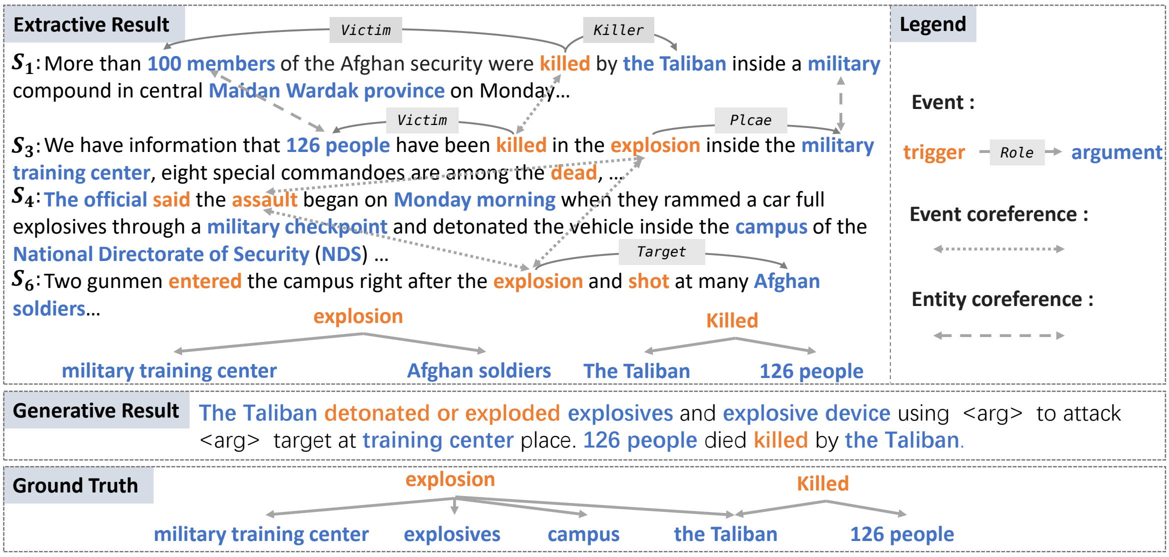
Extractive Result
Victim
Killer
S_1: More than 100 members of the Afghan security were killed by the Taliban inside a military compound in central Maidan Wardak province on Monday...
Victim
Plcae
S_3: We have information that 126 people have been killed in the explosion inside the military training center, eight special commandoes are among the dead, ...
S_4: The official said the assault began on Monday morning when they rammed a car full explosives through a military checkpoint and detonated the vehicle inside the campus of the National Directorate of Security (NDS) ...
Target
S_6: Two gunmen entered the campus right after the explosion and shot at many Afghan soldiers...
explosion
military training center
Afghan soldiers
Killed
The Taliban
126 people
Legend
Event :
trigger
Role
argument
Event coreference :
Entity coreference :
Generative Result
The Taliban detonated or exploded explosives and explosive device using <arg> to attack <arg> target at training center place. 126 people died killed by the Taliban.
Ground Truth
explosion
military training center
explosives
campus
the Taliban
Killed
126 people

Figure 4: A case studies to illustrate the effectiveness of the proposed extractive model and generative model.

Models	Coref F1	Infor F1
Extractive	38.41	26.96
-MultiLearn	-1.76	-1.81
-GlobalInter	-0.75	-0.48
-SchSamp	-2.23	-2.07

Table 5: Evaluation of ablation studies on the extractive model variants.

which indicates dropping the scheduled sampling strategy during training. The results are shown in Table 5 and we can observe that: (1) Multi-task learning can be benefit from joint learning for entity extraction, event extraction and pairwise classification, and we conjecture that multi-task learning can result in richer representation. (2) The introduction of the global-aware interaction can promote the interaction among triggers and entities, which contributes +0.62. (3) The scheduled sampling strategy, which alleviates the mismatch of entities and triggers for pairwise classification between training and inference, contributes greatly and improves the results by 2.15 F1 scores on average.

5.7 Case Studies

To visually show the effectiveness of the introduced two different solutions, we conduct case studies to compare the results of the extractive model and the generative model. As shown in Figure 4, we have the following observations: (1) With the extractive solution, we can get a detailed process of how to extract core events with informative arguments from a document. Firstly, the extraction model will predict a series of entities (color in blue) and triggers (color in orange) with their types. Then the extractive model connects events and arguments by event coreference and entity coreference. Furthermore, by core event detection, the model can filter out secondary events (i.e., the events triggered by "`entered`" and "`shot`") and result core structured events. (2) With the generative solution, we can get a core events description in a sequence formulation which can translate into structured events. (3) From the comparison of prediction results from the extractive model and the generative model, we can observe that the generative model performs better.

6 Conclusion and Future Work

In this paper, we explore focused and connected document-level event extraction. To achieve this, we annotate a new dataset, named WIKIEVENTS++, and introduce document-level evaluation metrics. Furthermore, we address this challenging task in two different manners and various experiments verify the effectiveness of the proposed methods. In this paper, we only focus on the entity coreference and event coreference to connect the events. But there are other connections between events, such as subevent relations, temporal relations and causal relations. In our future work, we will devote to exploring these connections to advance the study on document-level event extraction.

References

1992. *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *NIPS*.

Yee Seng Chan, Joshua Fasching, Haoling Qiu, and Bonan Min. 2019. Rapid customization for event extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 31–36, Florence, Italy. Association for Computational Linguistics.

Pei Chen, Hang Yang, Kang Liu, Ruihong Huang, Yubo Chen, Taifeng Wang, and Jun Zhao. 2020. 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, 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.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multipooling 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.

Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the most dominant event in a news article by mining event coreference relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 340–345, New Orleans, Louisiana. Association for Computational Linguistics.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, page 1. Lisbon.

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.

Xinya Du, Alexander Rush, and Claire Cardie. 2020. Document-level event-based extraction using generative template-filling transformers. *arXiv e-prints*, pages arXiv–2008.

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

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

Ralph Grishman and Beth Sundheim. 1996. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*.

Ralph Grishman, David Westbrook, and Adam Meyers. 2005. Nyu's english ace 2005 system description.

Felix Hamborg, Corinna Breitinger, and Bela Gipp. 2019. Giveme5w1h: A universal system for extracting main events from news articles. *arXiv preprint arXiv:1909.02766*.

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

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.

Ruihong Huang and Ellen Riloff. 2011. Peeling back the layers: detecting event role fillers in secondary contexts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1137–1147. Association for Computational Linguistics.

Ruihong Huang and Ellen Riloff. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 286–295. 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.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pretraining for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

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

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhengzhong Liu, Chenyan Xiong, Teruko Mitamura, and Eduard Hovy. 2018. Automatic event salience identification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1226–1236, 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.

Siddharth Patwardhan and Ellen Riloff. 2009. A unified model of phrasal and sentential evidence for information extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 151–160. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. brat: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

Kaiwen Wei, Xian Sun, Zequn Zhang, Jingyuan Zhang, Guo Zhi, and Li Jin. 2021. Trigger is not sufficient: Exploiting frame-aware knowledge for implicit event argument extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4672–4682, Online. Association for Computational Linguistics.

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

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.

Hang Yang, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021a. Multi-sentence argument linking via an event-aware hierarchical encoder. In *Proceedings of the 30th ACM International Conference on Information amp; Knowledge Management*, CIKM '21, page 3578–3582, New York, NY, USA. Association for Computing Machinery.

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

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

Xiaodong Yu, Wenpeng Yin, and Dan Roth. 2020. Paired representation learning for event and entity coreference. *arXiv preprint arXiv:2010.12808*.

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

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for Chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346, Hong Kong, China. Association for Computational Linguistics.

A Appendix

In the appendix, we incorporate the following details that are omitted in the main body due to the space limit.

- Section A.1 introduce the Hungarian Algorithm.
- Section A.2 show the hyper-parameter setting.

A.1 Evaluation for DEE

For considering event coreference and entity coreference during document-level evaluation for EE, we introduce two metrics: DocTri and DocArg.

A.1.1 DocTri and DocArg

DocTri considers trigger span with position, event type, and event coreference. Triggers in the same event coreference chain are clustered together. To match the predicted event clusters and the gold clusters, we adopt Kuhn–Munkres algorithm to get the optimal mapping. Then, according to the mapping results, we can calculate the Precision (P), Recall (R), and F1-measure (F1-score) for the matching and unmatched clusters. Similarly, DocTri considers argument span with position, role type, and entity coreference. Arguments in the same entity coreference chain are clustered together. Kuhn–Munkres algorithm is adopted to get the optimal mapping.

A.1.2 Kuhn-Munkres Algorithm

The Kuhn-Munkres Algorithm is a combinatorial optimization algorithm that solves the linear sum assignment problem. The linear sum assignment problem is also known as minimum weight matching in bipartite graphs. A problem instance is described by a matrix C, where each C[i,j] is the cost of matching vertex i of the first partite set and vertex j of the second set. The goal is to find a complete assignment of workers to jobs of minimal cost.

Formally, let X be a boolean matrix where $X[i,j]=1$ iff row i is assigned to column j. $C_{i,j}$ is the cost matrix of the bipartite graph. Then the optimal assignment has cost:

$$min \sum_i \sum_j C_{i,j} X_{i,j} \tag{6}$$

s.t. each row is assignment to at most one column, and each column to at most one row.

A.2 Hyper-parameter setting

Hyper-parameter	Value
Base encoder	Roberta-large
Base encoder-decoder	BART-large
Max sequence length for document	512
Max sequence length for sentence	128
Embedding size	1024
Hidden size	1024
Tagging scheme	BIO (Begin, Inside, Other)
Layers of Global Transformer	4
Optimizer	AdamW
Learning rate for Seq model	2e^{-5}
Learning rate for QA model	2e^{-5}
Learning rate for extractive model	2e^{-5}
Learning rate for generative model	1e^{-5}
Batch size	8
Dropout	0.1
Training epoch	50

Table 6: The hyper-parameter setting.