DocEE: A Large-Scale Dataset for Document-level Event Extraction

Anonymous ACL submission

Abstract

Event extraction (EE) is the task of identifying events and their types, along with the involved arguments. Despite the great success in sentence-level event extraction, events are more naturally presented in the form of document, with event arguments scattering in multiple sentences. However, a major barrier to promote document-level event extraction has been the lack of large-scale and practical training and evaluation datasets. In this paper, we present DocEE, a new document-level EE dataset including 20,000+ events, 100,000+ arguments. We highlight three features: large-scale annotations, fine-grained event arguments and application-oriented settings. Experiments show that even SOTA models show inferior performance on DocEE, especially in cross-domain settings, indicating that DocEE is still a challenging task. We will publish DocEE upon acceptance.

1 Introduction

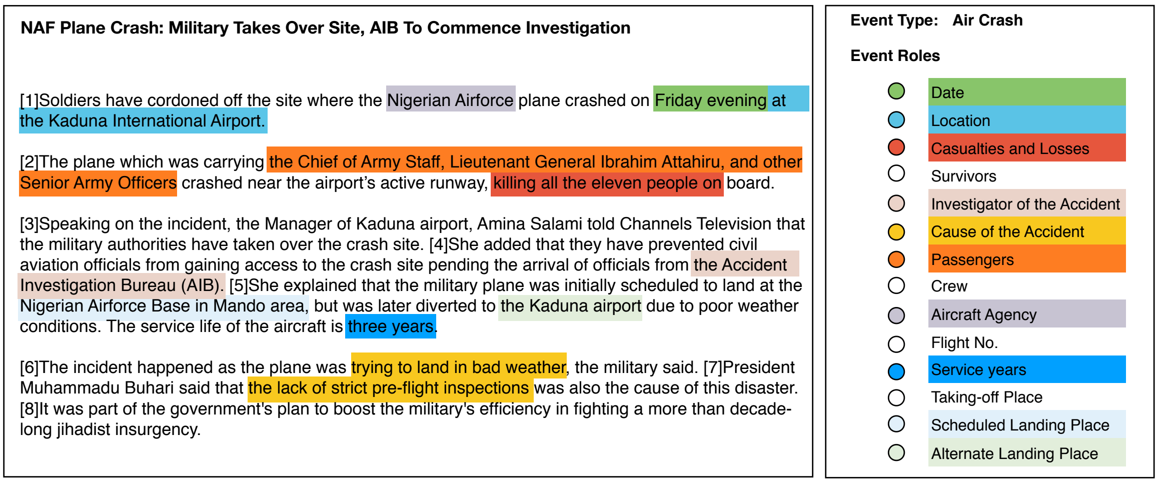
Event Extraction (EE) aims to detect an event from the text, disambiguate its semantic type from its event ontology, and also finds the event's arguments when they are expressed in text. EE is one of the fundamental tasks in text mining(Feldman and Sanger, 2006) and has many applications. For instance, it can monitor political or military crises to generate real-time notifications and alerts (Dragos, 2013), and dig the links and connections (e.g., Who Met Whom and When) between dignitaries for portrait analysis (Zhan et al., 2020).

Most existing datasets (ACE2005 and KBP2017) focus on sentence-level event extraction. They assume that both the event and the arguments involved in the same sentence. In recent years, various pre-training language models have been presented to improve the sentence-level EE and have achieved great success (Orr et al., 2018; Nguyen and Grishman, 2018; Tong et al., 2020b).

Despite these outstanding results, sentence-level EE is subject to inevitable limitations in practice. In reality, events usually appear in the form of documentary descriptions, and the arguments involved in the event are also scattered in various sentences (Hamborg et al., 2019). Taken Figure 1 as an example, the article mainly describes an *Air Crash* event, and in order to extract all arguments participated in *Air Crash*, one has to read the entire text and find answers from different sentences. For example, to extract the argument *Data*, we need to understand the first sentence, while to obtain the argument *Cause of the Accident*, we need to integrate information in sentence [6] and [7]. This process requires reasoning over multiple sentences, which is intuitively beyond the reach of sentence-level EE methods. Therefore, it is necessary to move EE forward from sentence level to document level.

In recent years, there are only a few datasets that focus on document-level event extraction. MUC-4(Grishman and Sundheim, 1996) consists of only 1700 news articles based on an ontology of 4 event types and 5 argument types. The arguments in MUC-4 lack type-oriented refinement and shared by all event types. WikiEvents(Li et al., 2021) consists of only 246 documents. Most of the event arguments (78%) in WikiEvents are still concentrated in a single sentence, and there are very few cross-sentence event arguments. RAMS(Ebner et al., 2020) limits the scope of the arguments in a 5-sentence window around its event trigger, which is not in line with the actual application, and the number of the argument types in RAMS is only 65, which is quite limited. In summary, existing datasets for document-level EE fail in the following aspects: small scale of data, insufficient refinement of event arguments and not application-oriented. It is urgent to develop a manually labeled, large-scale dataset to accelerate the research in document-level event extraction.

In the paper, we present DocEE, a large-scale



NAF Plane Crash: Military Takes Over Site, AIB To Commence Investigation

[1]Soldiers have cordoned off the site where the Nigerian Airforce plane crashed on Friday evening at the Kaduna International Airport.

[2]The plane which was carrying the Chief of Army Staff, Lieutenant General Ibrahim Attahiru, and other Senior Army Officers crashed near the airport's active runway, killing all the eleven people on board.

[3]Speaking on the incident, the Manager of Kaduna airport, Amina Salami told Channels Television that the military authorities have taken over the crash site. [4]She added that they have prevented civil aviation officials from gaining access to the crash site pending the arrival of officials from the Accident Investigation Bureau (AIB). [5]She explained that the military plane was initially scheduled to land at the Nigerian Airforce Base in Mando area, but was later diverted to the Kaduna airport due to poor weather conditions. The service life of the aircraft is three years.

[6]The incident happened as the plane was trying to land in bad weather, the military said. [7]President Muhammadu Buhari said that the lack of strict pre-flight inspections was also the cause of this disaster. [8]It was part of the government's plan to boost the military's efficiency in fighting a more than decade-long jihadist insurgency.

Event Type: Air Crash

Event Roles

● Date
● Location
● Casualties and Losses
○ Survivors
● Investigator of the Accident
● Cause of the Accident
● Passengers
○ Crew
● Aircraft Agency
○ Flight No.
● Service years
○ Taking-off Place
○ Scheduled Landing Place
○ Alternate Landing Place

Figure 1: A toy example in DocEE. The article mainly describes a *Air Crash* event. *Data*,*Location*, *Causality and Losses* and etc. are predefined argument types participating in the *Air Crash*, whose answers scatter in various sentences in the article.

human-annotated document-level EE dataset constructed from Wikipedia. Figure 1 illustrates a toy example of our DocEE. We construct DocEE with the following three features: 1) Large-scale. DocEE contains 21,450 document-level events with 109,395 arguments, far exceeding the scale of the existing document-level EE dataset. 2) More refined event arguments. DocEE has a total of 358 argument types, which is much more than the number of event argument types in existing dataset (5 in MUC-5 and 65 in RAMS). Besides the general arguments, we design more personalized event arguments for each event type. For instance, we specially design *Water Level* for *Flood* event and *Magnitude* for *Earthquake* event. 3) Towards more practical arguments extraction. The average number of sentences between arguments in DocEE is up to 10.1, which is much higher than the existing dataset (ACE2005 is 1 and MUC-4 is 4.0). With more scattered event arguments, our dataset can be more cope with realistic scenarios where the length of the article will be particularly long, and the argument of the event may appear in any corner of the article. Besides, we provide a good evaluation scenario for testing the ability of the pre-trained models that aim at handling long text understanding, such as longformer(Beltagy et al., 2020) and ∞-former(Martins et al., 2021).

To assess the challenges of DocEE, we implement 9 recent state-of-the-art EE models and test their capabilities in event classification and event argument extraction. Experiments show that even the performance of SOTA model is far lower than human performance, showing that the faintness of existing technology in processing document-level event extraction.

2 Related Datasets

Sentence-level Event Extraction Dataset Automatic Content Extraction(ACE2005 [1]) is a widely used sentence-level EE datasets, consisting of 599 documents with 8 event types and 33 subtypes (Tong et al., 2020a). Text Analysis Conference (TAC-KBP) [2] also releases three benchmarks: TAC-KBP 2015/2016/2017, with 9/8/8 event types and 38/18/18 event subtypes. RED[3] annotates events from 95 English newswires. Chinese Emergency Corpus (CEC) focuses on Chinese breaking news, with a total of 332 articles in 5 categories. MAVEN (Wang et al., 2020) and LSEE (Chen et al., 2017) only mark event trigger, with 168/21 types of event in 11,832/72,611 sentences. All of the above data sets either lack event argument annotations, or limit the event arguments to one sentence, failing to handle the situation where the event arguments are scattered in multiple sentences.

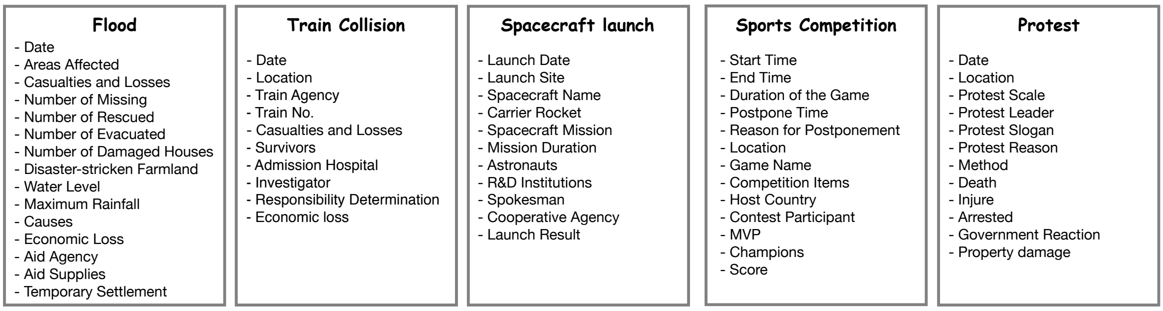
Document-level Event Extraction Dataset Most of the existing document-level event datasets only focus on event classification, but lack event argument labelings, such as 20news [4] and THUC-

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

[2] https://www.ldc.upenn.edu/collaborations/past-projects/tac-kbp

[3] https://catalog.ldc.upenn.edu/LDC2016T23

[4] https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups



Flood

- Date
- Areas Affected
- Casualties and Losses
- Number of Missing
- Number of Rescued
- Number of Evacuated
- Number of Damaged Houses
- Disaster-stricken Farmland
- Water Level
- Maximum Rainfall
- Causes
- Economic Loss
- Aid Agency
- Aid Supplies
- Temporary Settlement

Train Collision

- Date
- Location
- Train Agency
- Train No.
- Casualties and Losses
- Survivors
- Admission Hospital
- Investigator
- Responsibility Determination
- Economic loss

Spacecraft launch

- Launch Date
- Launch Site
- Spacecraft Name
- Carrier Rocket
- Spacecraft Mission
- Mission Duration
- Astronauts
- R&D Institutions
- Spokesman
- Cooperative Agency
- Launch Result

Sports Competition

- Start Time
- End Time
- Duration of the Game
- Postpone Time
- Reason for Postponement
- Location
- Game Name
- Competition Items
- Host Country
- Contest Participant
- MVP
- Champions
- Score

Protest

- Date
- Location
- Protest Scale
- Protest Leader
- Protest Slogan
- Protest Reason
- Method
- Death
- Injure
- Arrested
- Government Reaction
- Property damage

Figure 2: Examples of five events schema in DocEE.

News [5]. There are a few datasets annotated with cross-sentences event arguments. MUC-4 (Nguyen et al., 2016) only contains 4 event types and 5 argument types, and the 4 event types are close to each other and limited to the terrorist attack topic[6]. WikiEvents (Li et al., 2021) and RAMS(Ebner et al., 2020) consist of 246/9124 documents with only 59/65 argument types, and most of the arguments in the two datasets are shared among different event types without further refinement. In summary, these datasets either cover very few event types and argument types, or the data scale is quite limited, or the event argument is not carefully refined.

3 Constructing DocEE

Our main goal is to collect a large-scale dataset to promote the development of event extraction from sentence-level to document-level. In the following sections, we will first introduce how to construct the event schema, and then how to collect candidate data and how to label them through crowdsourcing.

3.1 Event Schema Construction

News is the first-hand source of hot events, so we focus on extracting events from news. Previous event schema, such as FrameNet (Baker, 2014) and HowNet (Dong and Dong, 2003), pays more attention to trivial actions such as *eating* and *sleeping*, and thus is not suitable for document-level news event extraction.

To construct schema, we gain insight from journalism. Journalism typically divides events into hard news and soft news (Reinemann et al., 2012; Tuchman, 1973). Hard news is an social emergency that must be reported immediately, such as earthquake, road accidents and armed conflict. Soft news refers to interesting incidents related to human life, such as celebrity deeds, sports events and other entertainment-centric reports. Based on the hard/soft news theory and the category framework in (Lehman-Wilzig and Seletzky, 2010), we define a total of 59 event types, with 31 hard news event types and 28 soft news event types. Detailed information is shown in Appendix Table 1. Our schema covers influential events of human concern, such as earthquake, floods and diplomatic summits, which cannot be extracted at the sentence level and require multiple sentences to describe.

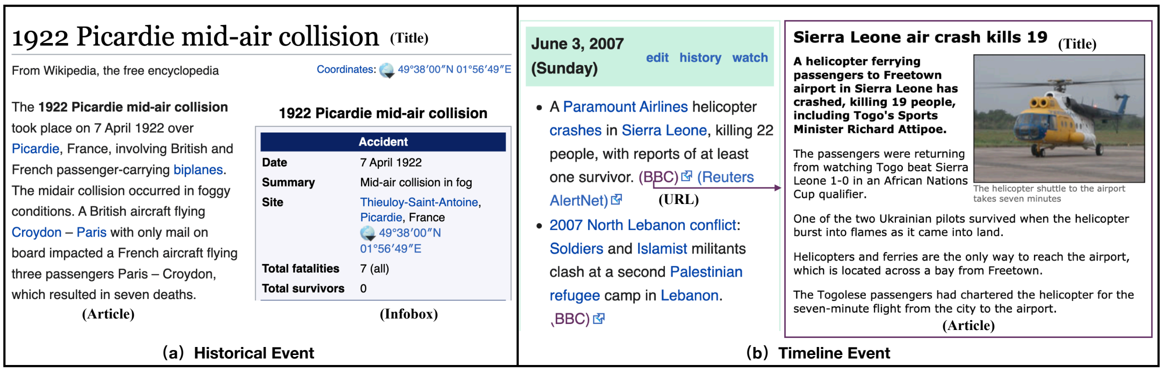
To obtain refined event arguments of each event type, we leverage infobox in Wikipedia. As shown in Figure (a) 3, the wiki page describes an event, and the keys in the infobox, such as *Date* and *Total fatalities*, can be regarded as the prototype arguments of the event. Based on this observation, we manually collect 20 wiki pages for each event type, and use their shared keys in infobox as our basic set of event arguments. After that, we invite 5 students to come up with more candidate event arguments to supplement the basic set. Finally, candidate event arguments with more than 3 votes will be accepted to our final event arguments schema. In total, we define 358 event arguments for 59 event type. On average, there are 5.1 event arguments per class. Figure 2 illiterates some examples of event arguments we defined. The complete event schema and corresponding examples can be found *Event Schema.md* in the supplementary materials.

3.2 Candidate Data Collection

In the section, we introduce how to collect candidate document-level events from Wikipedia. Wikipedia contains two kinds of events: historical events and timeline events (Hienert and Luciano, 2012). Examples of both are shown in Figure 3.

[5] http://thuctc.thunlp.org

[6] https://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html



1922 Picardie mid-air collision

From Wikipedia, the free encyclopedia

Coordinates: 49°38′00″N 01°56′49″E

The **1922 Picardie mid-air collision** took place on 7 April 1922 over Picardie, France, involving British and French passenger-carrying biplanes. The midair collision occurred in foggy conditions. A British aircraft flying Croydon – Paris with only mail on board impacted a French aircraft flying three passengers Paris – Croydon, which resulted in seven deaths.

1922 Picardie mid-air collision

Accident	
Date	7 April 1922
Summary	Mid-air collision in fog
Site	Thieuloy-Saint-Antoine, Picardie, France 49°38′00″N 01°56′49″E
Total fatalities	7 (all)
Total survivors	0

(a) Historical Event

June 3, 2007 (Sunday)

- A Paramount Airlines helicopter crashes in Sierra Leone, killing 22 people, with reports of at least one survivor. (BBC) (Reuters AlertNet)
- 2007 North Lebanon conflict: Soldiers and Islamist militants clash at a second Palestinian refugee camp in Lebanon. (BBC)

Sierra Leone air crash kills 19

A helicopter ferrying passengers to Freetown airport in Sierra Leone has crashed, killing 19 people, including Togo's Sports Minister Richard Attipoe.

The passengers were returning from watching Togo beat Sierra Leone 1-0 in an African Nations Cup qualifier.

The helicopter shuttle to the airport takes seven minutes

One of the two Ukrainian pilots survived when the helicopter burst into flames as it came into land.

Helicopters and ferries are the only way to reach the airport, which is located across a bay from Freetown.

The Togolese passengers had chartered the helicopter for the seven-minute flight from the city to the airport.

(b) Timeline Event

Figure 3: Two sources of candidate events in DocEE. The left is the historical event, which has its own wiki page, and the right are the timeline event, which are arranged according to the time unit by wiki editors. Each timeline event has a URL to the original news article.

Historical event refers to the event that has its own wiki page, such as *1922 Picardie mid-air collision* and *2014 Santos Cessna Citation accident*. Timeline events refer to events organized in chronological order in the wiki, such as *Current events/August 1998* [7] and *Current events/June 2010* [8], and each timeline event consists of a brief description and the URL of the original news. We adopt both kinds of events as our candidate data, because only using historical events will lead to uneven data distribution under our event schema, and timeline events can be a good supplement.

For historical event, we adopt wiki page as the document of the event argument to be annotated. For timeline event, we use the URL to download the original news article as the document of the event argument to be annotated. Noted that about half of the URLs in timeline event have invalid issues, so we use Scale SERP [9] to find alternative news on google and manually confirm their authenticity. For historical event, we adopt *templates+event type* as the query key to retrieve candidate events. The templates includes *"List of"+event type*, *event type+"in"+year*, *"Category:"+event type+"in"+country*, etc. For timeline event, we choose events between 1980 and 2021 as candidates, because there are few instances of events before 1980.

In order to balance the length of the article, we filtered out articles less than 5 sentences, and also truncated articles that were too long (more than 50 sentences). Finally, we select 44,000 candidate events from Wikipedia.

3.3 Crowdsourced Labeling

The crowdsourced labeling process consists of two stages.

3.3.1 Stage 1: Event Classification

At this stage, annotators are required to classify candidate events into predefined event types. Formally, given the candidate document-level event $e =< t, a >$, where t represents the title and a represents the article, Stage 1 aims to obtain label y for each e, where y belongs to the 59 event types defined in subsection 3.1. Instead of extracting as many events as possible, we focus on main event to grasp the key points, so Stage 1 is a single-label classification. Following (Hamborg et al., 2018), The main event refers to the event reflected in the title and mainly described in the article.

In total, we invite about 60 annotators to participate in Stage 1 annotation. A screenshot of the annotation page is shown in Figure 1 in Appendix. We first evaluate the quality of the annotators by proportional sampling inspection and weed out annotators with an accuracy rate of less than 70%. Then, each candidate event is annotated by two independent annotators. Once the results of the two annotators are inconsistent (32.8% in this case), a third annotator will be the final judge. If a candidate event does not belong to any predefined classes, we classify it into the other class, which accounts for 23.6% of the total data.

3.3.2 Stage 2: Event argument Extraction

At this stage, annotators are required to extract event arguments from the whole article. Formally, given the candidate event $e =< t, a >$, its event type y and the predefined event arguments R of event type y, Stage 2 aims to draw answers $Q_r =$

[7] https://en.wikipedia.org/wiki/Portal:Current_events/August 1998

[8] https://en.wikipedia.org/wiki/Portal:Current_events/June_2010

[9] https://app.scaleserp.com/playground

$\{start_r, end_r\}$ from the article a for each event argument $r \in R$.

Due to the heavy workload in Stage 2, we invite more than 90 annotators. A screenshot of the online annotation page is shown in Figure 2 in Appendix. According to our observations, the quality of annotated data is proportional to the number of rounds of revision review. Therefore, we use the *preliminary annotation - multiple rounds inspection* method for labeling. In the preliminary annotation step, each article will be labeled by an annotator. We distribute no more than two event types to each annotator in this step to make the annotators more focused. Then, in the step of multiple rounds inspection, we select high-precision annotators (44.4% of the total) to form a reviewer team, and each article will go through three rounds of quality inspection and error correction by three independent annotators in the reviewer team. In each round, the reviewer will check whether argument spans are correct and manually correct the wrong answers.

To make our event argument more informative, we train annotators to ensure that the answers to the event arguments are neither too general nor too wordy. For example, it is meaningless to answer *Aid Agency* with *some rescue agencies*. Without providing a specific organization name, the answer is too general, and users will not get any useful information after reading our extracted results. Also, it is too wordy to answer *Symptoms* with too delicate feelings, such as *he does not know where he is*. Just grasping the key points and answering *dizziness and confusion* is enough.

3.3.3 Remuneration

The annotators spend an average of 0.5 minutes labeling a piece of data in Stage 1, so we pay them 0.1$ for each piece of data. It takes about 5 minutes to label a piece of data in Stage 2 , so we pay 0.8$ for each piece of data.

4 Data Analysis of DocEE

In the section, we analyze various aspects of DocEE to provide a deep understanding of the dataset and the task of document-level event extraction.

4.1 Overall Statistic

In total, DocEE labels 21,450 valid document-level events and 109,395 event arguments. Each article is annotated with 5.1 event arguments on average. Event *Flood* has the highest average number of event arguments per article (11.8), while event *Join in an Organization* has the lowest average number of event arguments per article (3.1).

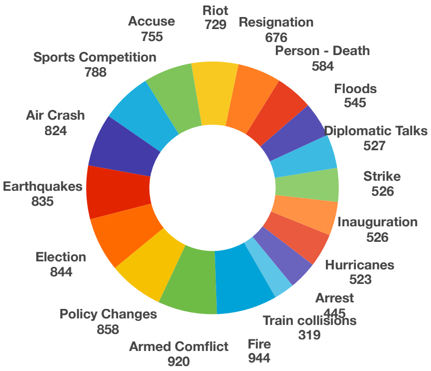


Figure 4: Top 18 event types in DocEE.

We compare DocEE to various representative event extraction datasets in Table 1, including sentence-level EE datasets ACE2005, KBP and document-level EE dataset MUC-4, Wikievents. We find that DocEE is larger than existing datasets in many aspects, including the total number of documents and event argument instances. Compared to MUC-4, DocEE has far more number of event arguments (109,395 to 2,641). The reason is that among the 1,700 documents in MUC-4, 47.4% of articles are not labeled with any event argument, while DocEE guarantees that each article contains at least three event argument labels in crowdsourcing process, which greatly solves the problem of data scarcity of the event arguments in document-level event extraction.

4.2 Event Type Statistic

Figure 4 shows the distribution of the top 18 event types that have the most number of instances in DocEE. DocEE covers a variety of event types, including Fire (4.5%), Armed Conflict (4.4%), Policy Changes (4.1%), Election (4.0%), Earthquake (3.9%), Air Crash (3.9%), Sports Competition (3.7%), etc. The instance distribution is relatively even, where there are 27.1% of classes with more than 500 instances and 72.8% of classes with more than 200 instances. More detailed information is shown in Table 1 in Appendix.

4.3 Event argument Type Statistic

We randomly sampled 100 articles from dev and test set, which contain 571 event arguments in-

Table 1: Statistics of EE datasets (isDocEvent: whether the event in the corpus at the document-level, EventTyp.: event type, ArgTyp.: event argument type, Doc.: document, Sent.: sentence, ArgInst.: event arguments, ArgScat.: the number of sentences in which event arguments of the same event are scattered)

Datasets	#isDocEvent	#EventTyp.	#ArgTyp.	#Doc.	#Tok.	#Sent.	#ArgInst.	#ArgScat.
ACE2005	✗	33	35	599	290k	15,789	9,590	1
KBP2016	✗	18	20	169	94k	5,295	7,919	1
KBP2017	✗	18	20	167	86k	4,839	10,929	1
MUC-4	✓	4	5	1,700	495k	21,928	2,641	4.0
WikiEvents	✓	50	59	246	190k	8,544	5,536	2.2
RAMS	✓	139	65	9,124	957k	34,536	21,237	4.8
DocEE(ours)	✓	59	358	21,450	14,540k	658,626	109,395	10.4

stances, and manually analyze the event argument types.

We first classify event arguments into three categories according to the answer type, including single answer, multiple answers and repeatedly described answer. As shown in Table 2, 70% event arguments have the single answer, and 26% of the event arguments have multiple answers which should be extracted from different places in the article. With multiple-answer event arguments, DocEE poses a greater challenge to the model's recall capability. Besides, 4% event arguments are repeatedly described in the article. We label all answers to provide more supervision signals to facilitate model training.

Then, we classify event arguments based on the length of the answers. 52% event arguments are no more than 3 words, and most of them are named entities such as people, time and location. While 40% event arguments are between 4 and 10 words and 8% event arguments are answered by more than 10 words, such event arguments mainly include *Cause of the Accident*, *Investigation Results* , etc.

5 Experiments on DocEE

5.1 Benchmark Settings

We design two benchmark settings for evaluation: normal setting and cross-domain setting. In the normal setting, we hope the training set and test set to be identically distributed. Specifically, for each class, we randomly select 80% of the data as the training set, 10% of the data as the validation set, and the remaining 10% of the data as the test set.

In order to be application-oriented, we design cross-domain setting to test the transfer capability of the SOTA models. We choose the event type under the subject of natural disasters as the target domain, including Floods, Droughts, Earthquakes, Insect Disaster, Famine, Tsunamis, Mudslides, Hurricanes, Fire and Volcano Eruption, and adopt the remaining 49 event types as source domains. The division reduces the overlap of the event arguments between the source domain and the target domain. In this setting, the models will first be pre-trained on the source domain, and then conduct 5-shot fine-tuned on the target domain. The detailed data split for each setting is shown in Table 3.

5.2 Hyperparameters

We use base model for all the transformer-based methods, and set the learning rate to 2e-5. The batch size is 128 and the maximum document length is 512. All baselines are implemented by HuggingFace [10], and all models can be fit into eight V100 GPUs with 16G memory. The training procedure lasts for about a few hours. For all the experiments, we report the average result of five runs as the final result. In human evaluation, we randomly select 1000 document-level events and invite three students to label them. The final result is the average of their labeling accuracy.

5.3 Event Classification

5.3.1 Baselines

We adopt CNN-based method and various transformer-based methods as our baselines, including: 1) **TextCNN** (Kim, 2014) uses different sizes CNN kernels to extract key information in text for classification. 2) **BERT** (Devlin et al., 2018) exploits the unsupervised objective functions for pre-training, including masking language model (MLM) and next sentence prediction. 3) **ALBERT** (Lan et al., 2020) proposes a self-supervised loss to improve inter-sentence coherence in BERT. 4) **DistillBert** (Sanh et al., 2019) combines language modeling, knowledge distillation and cosine-distance losses to improve the pre-training perfor-

[10] https://huggingface.co/models

Table 2: Answer types of event arguments in DocEE.

Answer Types	**%**	**Examples**
Single Answer	70	A masked man in a black hoodie showed a **gun** and was handed money before running east on Warren Street, according to the initial report. Event Type: Bank Robbery Event argument: Weapon Used
Multiple Answers	26	At around 6:20 a.m. **a lorry**, driven by David Fairclough of Wednesfield, rammed into the rear of **a tanker**, which then struck **a car** in front and exploded. The ensuing pile-up involved **160 vehicles** on a 400-yard (370 m) stretch of the motorway. Event Type: Road Crash Event argument: Number of Vehicles involved in the Crash
Repeatedly Described	4	The 2009 **Bank of Ireland** robbery was a was the largest bank robbery in the Republic of Ireland's history. Criminals engaged in the tiger kidnapping of a junior bank employee in the College Green cash centre of the **Bank of Ireland** in Dublin, Ireland, on 27 February 2009. Event Type: Bank Robbery Event argument: Bank Name

Table 3: Data split in the normal setting and cross-domain setting. #Typ. event type, #Doc. document, #ArgInst. event arguments

Method	Normal			Cross-Domain		
	Train	Dev	Test	Train	Dev	Test
#Typ.	59	59	59	59	10	10
#Doc.	15.9k	2740	2772	12.7k	158	164
#ArgInst.	74.2k	10k	10k	65.0k	776	848

mance. 5) **RoBERTa** (Liu et al., 2019) builds on BERT and modifies key hyperparameters, removing the next-sentence pretraining objective and training with much larger mini-batches and learning rates. Following (Kowsari et al., 2019), we use Precision(P), Recall(R) and F1 score as the evaluation metrics. We report the macro averaging to avoid overestimation caused by classes with more examples.

Table 4: Overall Performance on Event Classification(%).

Method	Normal Setting			Cross-Domain Setting		
	P	R	F	P	R	F
TextCNN	53.3	49.2	51.2	0.4	1.7	0.6
BERT	67.5	65.9	65.5	24.4	25.6	23.2
ALBERT	63.0	59.6	59.8	19.9	18.8	16.3
DistilBert	70.5	67.2	67.1	22.3	18.5	18.6
RoBERTa	70.1	68.7	68.2	24.8	24.0	23.4
Human	91.4	94.7	92.7	-	-	-

5.3.2 Overall Performance

Table 4 shows the experimental results under the normal and cross-domain settings, from which we have the following observations: 1) Compared with CNN based model (TextCNN), the transformer based models (BERT, ALBERT, DistillBert, RoBERTa) perform better. This is because the transformer based models are pre-trained on a large-scale unsupervised corpus and have more background semantic knowledge to rely on when judging event types. 2) Humans have achieved high scores on DocEE, verifying the high quality of our annotated data sets. 3) There is still a big gap between the performance of the current SOTA models and human beings, which indicates that more technological advances are needed in future work. Human can connect and merge key information to form a knowledge network to help them understand the main event, while deep learning models typically fail in long text perception. 3) There is a significant performance degradation from the normal setting to the cross-domain setting, which shows that domain migration is still a huge challenge for current SOTA models. Among them, DistillBert's performance drops the most. The reason may be that the parameter scale in DistillBert is relatively small, and the reserved source domain knowledge is limited.

5.4 Event argument Extraction

5.4.1 Baselines

We introduce three kinds of mainstream baselines for evaluation: 1) Sequence Labeling Methods **BERT-Seq**. This method uses the pre-trained BERT model to sequentially label words in the article. Given the input article $A = \{w_1, w_2, \dots, w_n\}$, the output of Sequence Labeling Methods is $O = \{y_1, y_2, \dots, y_n\}$, where $y_i \in R$ and R is the set of the event argument types. 2) Q&A Methods **BERT-QA**. This method uses the event argument as the question to query the article for answer. Given the input article $a = \{w_1, w_2, \dots, w_n\}$, the event arguments $R = \{r_1, r_2, \dots, r_m\}$ as the question, the output is $O = \{< start_{r_1}, < end_{r_1} >, < start_{r_2}, < end_{r_2} >, \dots, < start_{r_m}, < end_{r_m} > \}$. We give -1 for these not mentioned event argu-

Table 5: Overall Performance on Event argument Extraction(%).

Methods	Normal Setting						Cross-domain Setting					
	EM			HM			EM			HM		
	P	R	F	P	R	F	P	R	F	P	R	F
BERT-Seq(sent)	68.3	24.7	34.5	71.5	28.1	36.2	32.4	10.3	18.6	34.7	10.8	19.2
BERT-Seq(chunk)	71.0	29.9	40.1	74.2	31.3	42.3	36.3	13.8	21.4	37.6	14.4	24.0
BERT-Seq(doc)	69.1	33.5	43.2	73.8	34.9	45.4	38.8	18.6	25.3	40.0	19.1	26.2
BERT-QA(chunk)	60.4	33.1	38.9	62.7	35.8	40.6	25.6	14.0	16.8	29.1	13.4	17.6
BART-Gen(chunk)	55.7	34.2	36.8	59.3	36.3	39.1	27.6	13.3	16.2	28.8	13.6	17.9
Human	87.81	84.20	85.96	80.94	87.16	89.0	-	-	-	-	-	-

ments. 3) Generative Methods **BART**. This method leverage the generative transformer-based encoder-decoder framework (BART) to directly generate the event arguments from the article. Given the input article $a = \{w_1, w_2, \dots, w_n\}$, the event arguments $R = \{r_1, r_2, \dots, r_m\}$, the output is O={The r_1 of t is o_1. The r_2 is of t is o_2, ... The r_m is of t is o_m. }. Considering the length limitation of pre-trained models, we split the article in three different ways. **(Sent)** means to split the article by sentence [11]. **(Chunk)** means to split the article by every 256 tokens. **(Doc)** means no splitting. We adopt Longformer(Beltagy et al., 2020) in *(doc)* situation. The longest article in DocEE contains about 7000 tokens, and the Longformer can still load the entire article at once.

Following the prior work (Du and Cardie, 2020), we use Head noun phrase Match (HM) and Exact Match (EM) as two evaluation metrics. HM is a relatively relaxed metric. As long as the head noun of the predicted result is consistent with the golden label, it will be judged as correct. While EM requires that the prediction result is exactly the same as the gold label, which is relatively stricter.

5.4.2 Overall Performance

As shown in Table 5, all the three kinds of models show inferior performance on the task of event argument extraction. The best performances of the sequential labeling model BERT-Seq, question answering model BERT-QA and generation model BERT-Gen, only reach 43.2%, 38.9% and 36.8% in the exact matching of F1 score.

One possible reason is that existing models cannot handle long texts well. Compared with traditional information extraction tasks, such as NER and relation extraction, Document-level EE(our task) poses a higher challenge to the model's ability to process long texts: the model has to read the entire text before determining the argument type of a span. Although a few models have been proposed to improve the long text capabilities of pre-trained models (such as longformer), and have achieved good results, (the performance of long-former (BERT-seq(doc)) is superior to BERT-seq(sent) and BERT-seq(The chunk) as shown in Table 5), but these models still have a big performance gap compared with human beings.

Another reason is the difficulty of semantic understanding, which is reflected in two aspects: 1) Event argument Confusion. EE models often confuse the argument of the main event with the arguments of other events mentioned in the article. For instance, the article mainly describes *the 2021 U.S. Alaska Peninsula earthquake*, and also mentions other historical earthquakes, saying that *The Wenchuan earthquake occurred in 2008 has the same magnitude as the Alaska Peninsula earthquake*. When asking the *Date* of the main event, the model often fails to understand the right answer is 2021, not 2008. 2) Over-labeling. EE models often mistake unrelated entities as event arguments. For example, when extracting the event argument *Attack Target* in the *the 911 terrorist attack on the Pentagon* event, except to the correct answer *the New York Pentagon*, EE models often mistake other unrelated location entities in the article (such as *Mount Sinai Hospital*) as one of the answers.

6 Conclusion

In this paper, we present DocEE, a large-scale document-level EE dataset to promote event extraction from sentence-level to document-level. Comparing to existing datasets, DocEE greatly expands the data scale, with more than 20,000 events and 100,000 argument, and contains more refined event arguments. Experiments show that even for the SOTA models, DocEE remains an open issue.

[11] https://www.nltk.org/api/nltk.tokenize.html

References

Collin F. 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.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

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.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Zhendong Dong and Qiang Dong. 2003. Hownet - a hybrid language and knowledge resource. In *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, pages 820–824.

Valentina Dragos. 2013. Developing a core ontology to improve military intelligence analysis. *International Journal of Knowledge-based and Intelligent Engineering Systems*, 17(1):29–36.

Xinya Du and Claire Cardie. 2020. 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.

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.

Ronen Feldman and James Sanger. 2006. *Information Extraction*, page 94–130. Cambridge University Press.

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*.

Felix Hamborg, Corinna Breitinger, and Bela Gipp. 2019. Giveme5w1h: A universal system for extracting main events from news articles. In *Proceedings of the 13th ACM Conference on Recommender Systems, 7th International Workshop on News Recommendation and Analytics (INRA 2019)*.

Felix Hamborg, Soeren Lachnit, Moritz Schubotz, Thomas Hepp, and Bela Gipp. 2018. Giveme5w: Main event retrieval from news articles by extraction of the five journalistic w questions.

Daniel Hienert and Francesco Luciano. 2012. Extraction of historical events from wikipedia. *CoRR*, abs/1205.4138.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882.

Kowsari, Jafari Meimandi, Heidarysafa, Mendu, Barnes, and Brown. 2019. Text classification algorithms: A survey. *Information*, 10(4):150.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations.

Sam N. Lehman-Wilzig and Michal Seletzky. 2010. Hard news, soft news, 'general' news: The necessity and utility of an intermediate classification. *Journalism*, 11(1):37–56.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. *CoRR*, abs/2104.05919.

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.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2021. ∞-former: Infinite memory transformer.

Kiem-Hieu Nguyen, Xavier Tannier, Olivier Ferret, and Romaric Besançon. 2016. A dataset for open event extraction in English. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1939–1943, Portorož, Slovenia. European Language Resources Association (ELRA).

T. Nguyen and R. Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Walker Orr, Prasad Tadepalli, and Xiaoli Fern. 2018. Event detection with neural networks: A rigorous empirical evaluation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 999–1004, Brussels, Belgium. Association for Computational Linguistics.

Carsten Reinemann, James Stanyer, Sebastian Scherr, and Guido Legnante. 2012. Hard and soft news: A review of concepts, operationalizations and key findings. *Journalism*, 6(2):221–239.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR*, abs/1910.01108.

Meihan Tong, Shuai Wang, Yixin Cao, Bin Xu, Juanzi Li, Lei Hou, and Tat-Seng Chua. 2020a. Image enhanced event detection in news articles. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9040–9047.

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020b. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5887–5897, Online. Association for Computational Linguistics.

Gaye Tuchman. 1973. Making news by doing work: Routinizing the unexpected. *American journal of Sociology*, 79(1):110–131.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A massive general domain event detection dataset. *CoRR*, abs/2004.13590.

Ge Zhan, Ming Wang, and Meiyi Zhan. 2020. Public opinion detection in an online lending forum: Sentiment analysis and data visualization. In *2020 IEEE 5th International Conference on Cloud Computing and Big Data Analytics (ICCCBDA)*, pages 211–213.