DocEE: A Large-Scale and Fine-grained Benchmark for Document-level Event Extraction

Anonymous ACL submission

Abstract

Event extraction aims to identify an event and then extract the arguments participating in the event. Despite the great success in sentence-level event extraction, events are more naturally presented in the form of documents, 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 event extraction dataset including 20,000+ events, 100,000+ arguments. We highlight three features: large-scale manual annotations, fine-grained argument types and application-oriented settings. Experiments show that there is still a big gap between state-of-the-art models and human beings (43% Vs 85% in F1 score), indicating that DocEE is an open issue. We will publish DocEE upon acceptance.

1 Introduction

Event Extraction (EE) aims to detect events from text, including event classification and event argument extraction. 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 (e.g., ACE2005 [1] and KBP2017 [2]) focus on sentence-level event extraction, while events are usually described at the document level, and event arguments are typically scattered across difference sentences (Hamborg et al., 2019). Figure 1 shows an *Air Crash* event. To extract argument *Data*, we need to read sentence [1], while to extract argument *Cause of the Accident*, we need to integrate information in sentence [6] and [7]. Clearly, this requires reasoning over multiple sentences and modeling long-distance dependency, intuitively beyond the reach of sentence-level EE. Therefore, it is necessary to move EE forward from sentence-level to document-level.

Only a few datasets are curated for document-level EE. MUC-4(Grishman and Sundheim, 1996) provides 1,700 news articles annotated with 4 event types and 5 argument types. The 5 arguments are shared among different event types without further refinement. WikiEvents(Li et al., 2021) consists of only 246 documents with very few (22% of total) cross-sentences argument annotations. 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. Doc2EDAG, TDJEE and GIT (Zheng et al., 2019; Wang et al., 2021; Xu et al., 2021) contain only 5 event types and 35 argument types in financial domain. In summary, existing datasets for document-level EE fail in the following aspects: small scale of data, limited coverage of domain and insufficient refinement of argument types. Therefore, 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 human-annotated document-level EE dataset. Figure 1 illustrates an example of DocEE. We highlight the following three contributions of DocEE to this field: 1) Large-scale Manual Annotations. DocEE contains 21,450 document-level events with 109,395 arguments, far exceeding the scale of the existing document-level EE dataset. The large-scale annotations of DocEE can provide sufficient training and testing data, to fairly evaluate EE models. 2) Fine-grained argument types. DocEE has a

[1] https://catalog.ldc.upenn.edu/LDC2006T06
[2] https://tac.nist.gov/2017/KBP/

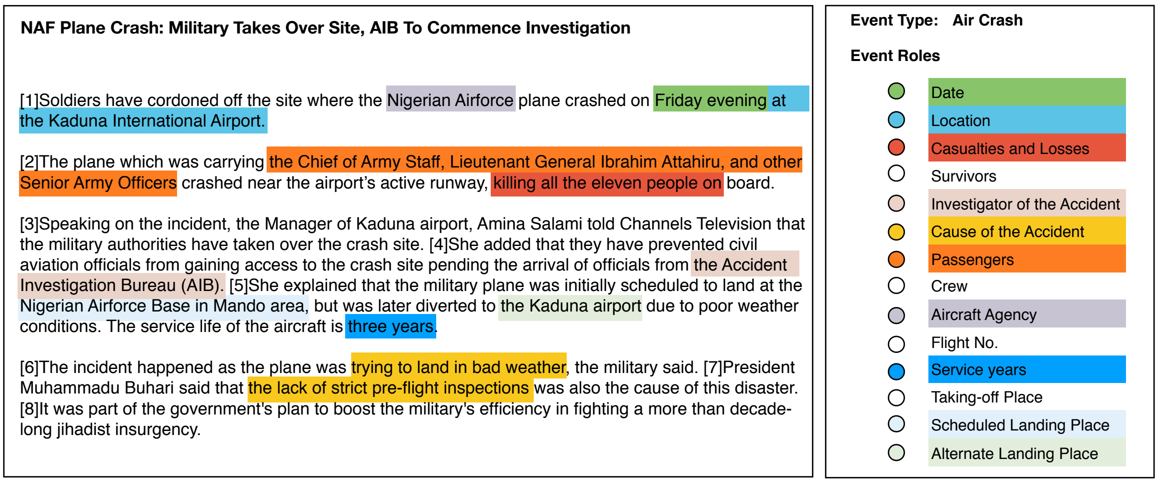

Figure 1: An example from DocEE. Each document in DocEE is annotated with event type and involved event arguments. In the example, the document mainly describes a *Air Crash* event which contains the following arguments: *Data*,*Location*, *Causality and Losses* and etc. We use different colors to distinguish event arguments.

total of 358 argument types, which is much more than the number of argument types in existing dataset (5 in MUC-5 and 65 in RAMS). Besides the general arguments, such as time and location, we design more personalized event arguments for each event type, such as *Water Level* for *Flood* event and *Magnitude* for *Earthquake* event. These fine-grained roles can bring more detailed semantics and deeper understanding of the documents. 3) Application-oriented settings. In the actual application, event extraction often face the problems that how to quickly adapt from the rich-resource domains to new domains. Therefore, we have added a cross-domain setting to better test the transfer capability of the EE models. In addition, unlike RAMS, DocEE removes the limitation that the arguments range should be within a certain window, to better 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.

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] consists of 599 documents with 8 event types and 33 subtypes. 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 annotate event triggers, with 168/21 types of trigger instances in 11,832/72,611 sentences. Based on them, various pre-training language models have been proposed to improve the sentence-level EE and have achieved great success (Orr et al., 2018; Nguyen and Grishman, 2018; Tong et al., 2020).

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 THUCNews [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 246/9,124 documents with

[3]https://catalog.ldc.upenn.edu/LDC2016T23
[4]https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups
[5]http://thuctc.thunlp.org
[6]https://www-nlpir.nist.gov/related_projects/muc/muc_data/muc_data_index.html

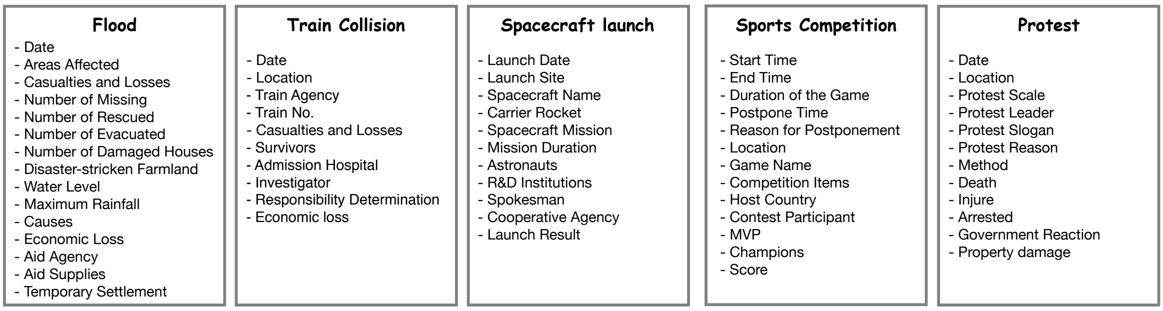

Figure 2: Five examples of event schema in DocEE.

only 59/65 argument types, and most of the arguments in the two datasets are shared among different event types without further refinement. Doc2EDAG, TDJEE and GIT (Zheng et al., 2019; Wang et al., 2021; Xu et al., 2021) only define 5 event types and 35 argument types in financial domain. In summary, these datasets either cover very few event 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 event 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 construct argument schema, 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 argument types. After that, we further expand the basic set. Specifically, for event type e, we first collect 20 news from New York Times, and then invited 5 students (native English-speaking, major in journalism) to summarize the key facts the public would like to learn from the news of e. For instance, in *Flood* event news, *Water Level* is a key fact, because it is an important factual basis for flood cause analysis and disaster relief decision-making, and can arouse widespread concern. Finally, by merging the key facts of the 5 students, we complete the argument types expansion. To ensure the quality, we further invite the above 5 students to make a trial labeling on the collected news, and filter argument types that appear less frequently in the article.

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. We choose wiki as

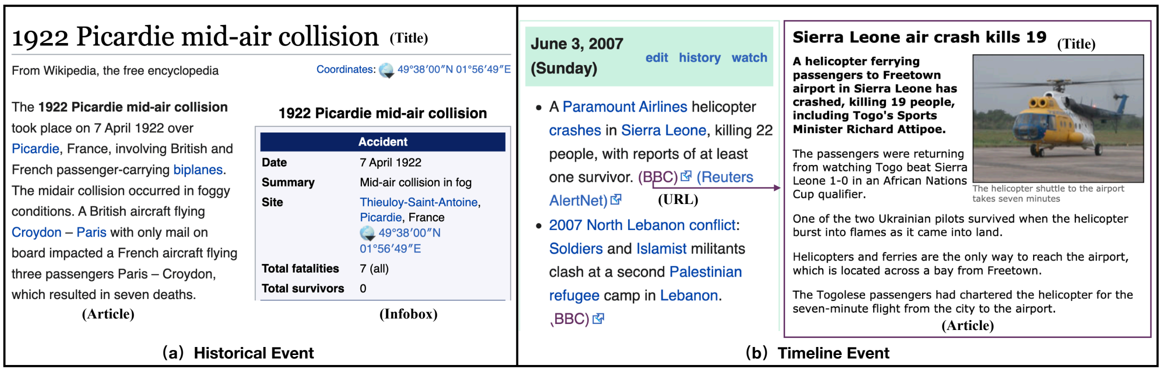

Figure 3: Two sources of candidate events in DocEE. The left is a historical event, which has its own wiki page, and the right are two timeline events arranged in a wiki page by time unit. Each timeline event consists of a brief description and a URL pointed to original news.

our annotation source. Wiki contains two kinds of events: historical events and timeline events (Hienert and Luciano, 2012). Historical event refers to the event that has its own wiki page, such as *1922 Picardie mid-air collision*. Timeline events refer to the news events that organized in chronological order, such as *A heatwave strikes India and South Asia* in wiki page *Portal:Current_events/June_2010* [7]. Figure 3 shows examples of two events. 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 [8] 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

Given the candidate events and the predefined event schema, we now introduce how to annotate them through crowdsourcing. 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. Following (Nguyen et al., 2016; Zheng et al., 2019), we adopt a no-trigger-words design. Following (Hamborg et al., 2018; Hsi, 2018), we focus on main event classification, so Stage 1 is a single-label classification task. Specifically, the main event refers to the event reflected in the title and mainly described in the article. Formally, given the candidate 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.

In total, we invite about 60 annotators to participate in Stage 1 annotation. The online annotation page is displayed in Figure 1 in Appendix. We first manually label 100 articles as standard answers to *pre-test* annotators, and weed out annotators with an accuracy rate of less than 70%, which left us 48 valid annotators. Then, we ask two independent annotators to annotate each candidate event. 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.

[7] en.wikipedia.org/wiki/Portal:Current_events/June_2010
[8] https://app.scaleserp.com/playground

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 argument types R of y, Stage 2 aims to find all the arguments from the article a.

Due to the heavy workload in Stage 2, we invite more than 90 annotators. An example of the online annotation page is shown in Figure 2 in Appendix. We use *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 via sampling inspection to form a reviewer team (44.4% of the total), and each article will go through three rounds of error correction by three independent annotators in the reviewer team. After each round, we randomly check 100 pieces of data, and find that the accuracy rate has steadily increased from 26.35%, 56.24%, 76.83% to 85.96%, which shows the effectiveness of our labeling method.

For event argument with multiple mentions in the document, for example, *Cause of the Accident* in Figure 1 has two mentions, we will label all mentions to ensure the completeness of the extraction. Repeated mentions will only be labeled once to reduce the burden on the annotator. Noted that we will not label the mentions that just simply repeat the argument type name, for example, to answer *Aid Agency* with *some rescue agencies*, to prevent the mention from being too general.

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).

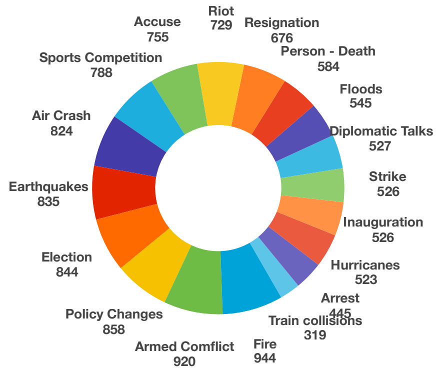

Figure 4: Top 18 event types in DocEE.

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, RAMS. We find that DocEE is larger than existing datasets in many aspects, including the documents numbers and argument instances numbers. 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.

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

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)

4.3 Event Arguments Statistic

We randomly sample 100 articles from DocEE for manual analysis, which contains a total of 571 event arguments instances.

We first classify event arguments based on their mention numbers. As shown in Table 2, 70% event arguments have unique mention, and 30% event arguments have multiple mentions, which poses a greater challenge to the model's recall capability. Then, we classify event arguments based on their mentions length. 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 event type, 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 argument types 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 [9], 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 masking language model (MLM) and next sentence prediction for pre-training. 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 BERT. 5) **RoBERTa** (Liu et al., 2019) builds on BERT and trains 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

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

Categories	%	Examples
Argument with Unique Mention	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
Argument with Multiple Mentions	30	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

Table 2: Statistical analysis of event arguments in DocEE.

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

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

macro averaging to avoid overestimation caused by classes with more examples.

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	-	-	-

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

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 TextCNN, transformer based models (BERT, ALBERT, DistillBert, RoBERTa) perform better, which are pre-trained on a large-scale unsupervised corpus and have more background semantic knowledge to rely on. 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. 4) 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 four kinds of mainstream baselines for evaluation: 1) Sequence Labeling Methods. **BERT-Seq** uses the pre-trained BERT model to sequentially label words in the article. Given the input article $A = \{w_1, w_2, \ldots, w_n\}$, the output of Sequence Labeling Methods is $O = \{r_1, r_2, \ldots, r_n\}$, where $r \in R$ and R is the set of the argument types. 2) Q&A Methods. **BERT-QA** uses the argument type as question to query the article for answer. Given the input article A, the argument type $r \in R$ as the question, the output is $O = \{start_r, end_r\}$. We give -1 for these not mentioned event arguments. **Ontology-QA**. Following (Vargas-Vera and Motta, 2004), we refine the initial query in BERT-QA with argument ontology knowledge obtained from Oxford dictionary (Dictionary, 1989). 3) Generative Methods. **BART-Gen**(Yan et al., 2021) leverages the generative transformer-based encoder-decoder framework (BART) to directly generate arguments from the article. Given the input article A, the argument types $R = \{r_1, r_2, \ldots, r_m\}$, the output is $O = \{start_{r_1}, end_{r_1}, r_1, start_{r_2}, end_{r_2}, r_2, \ldots, start_{r_m}, end_{r_m}, r_m\}$. 4) Task-specific Methods. **DocEDAG**(Zheng et al., 2019) generates an entity-based directed acyclic graph for document-level EE. **MG-Reader** (Du and Cardie, 2020a) improves document-level EE by proposing a novel multi-granularity reader to dynamically aggregate information in sentence and paragraph-level. The imple-

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
Ontology-QA(chunk)	69.6	30.9	39.8	73.2	33.1	43	38.3	14.5	22.9	38.9	15	24.6
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
Doc2EDAG(chunk)	68.5	30.3	38.4	69.2	31.5	39.5	35.2	11.3	20.1	35.2	11.7	20.8
MG-Reader(seq+chunk)	69.3	30.1	38.2	72.6	31.8	41.7	36.2	12.9	20.7	37.1	13.8	22.7
Human	87.8	84.2	85.9	80.9	87.2	89.0	-	-	-	-	-	-

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

mentation of the two baselines follows the original paper [10][11]. Considering the length limitation of pre-trained models, we split the article in three different ways. **(Sent)** means to split the article by sentence [12]. **(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, 2020b), 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, there is a big gap between the performance of SOTA models and human performance (43.2% Vs 85.9% in F score), indicating that document-level event argument extraction remain a challenge task.

The failure of existing baselines may be due to two reasons. One possible reason is the catastrophic forgetting in neural networks. Compared to NER and sentence-level EE, document-level EE(our task) highlights the model's capability 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), BERT-seq(chunk) and MG-reader as shown in Table 5), but these models still have a big performance gap compared with human beings.

Another reason is the inferior capability in semantic understanding, which is reflected in two aspects: 1) EE models fail to distinguish arguments of similar events. For instance, the article mainly describes *the 2021 U.S. Alaska Peninsula earthquake*, and also briefly mentions *2008 Wenchuan earthquake*. When asking the *Date* of the main event, EE models are easy to confuse the correct answer 2021 with the wrong answer 2008. 2) EE models often mistake unrelated entities for 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.

We believe that the following research directions are worthy of attention: 1) Exploring pre-trained models with stronger long text processing capabilities. 2) Exploiting ontology and commonsense knowledge to improve the semantic understanding of EE models.

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.

[10] https://github.com/dolphin-zs/Doc2EDAG
[11] https://github.com/xinyadu/doc_event_role
[12] 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.

Oxford English Dictionary. 1989. Oxford english dictionary. *Simpson, Ja & Weiner, Esc*.

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. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. *CoRR*, abs/2005.06579.

Xinya Du and Claire Cardie. 2020b. 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.

Andrew Hsi. 2018. *Event Extraction for Document-Level Structured Summarization*. Ph.D. thesis, Carnegie Mellon University.

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.

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, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. 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.

Maria Vargas-Vera and Enrico Motta. 2004. Aqua–ontology-based question answering system. In *Mexican International Conference on Artificial Intelligence*, pages 468–477. Springer.

Peng Wang, Zhenkai Deng, and Ruilong Cui. 2021. Tdjee: A document-level joint model for financial event extraction. *Electronics*, 10(7):824.

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.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. *CoRR*, abs/2105.14924.

Hang Yan, Tao Gui, Junqi Dai, Qipeng Guo, Zheng Zhang, and Xipeng Qiu. 2021. A unified generative framework for various ner subtasks. *arXiv preprint arXiv:2106.01223*.

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.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2edag: An end-to-end document-level framework for chinese financial event extraction. *arXiv preprint arXiv:1904.07535*.