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
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. She explained that the military plane was initially scheduled to land at the Kaduna International Airport.

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 further stated 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. President Muhammadu Buhari said that the lack of strict pre-flight inspections was also the cause of this disaster.

7. It was part of the government’s plan to boost the military’s efficiency in fighting a more than decade-long jihadist insurgency.

8. 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.

9. 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-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 246/9,124 documents with

1. 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 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.

2. 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 THUC-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 246/9,124 documents with

---

1 https://catalog.ldc.upenn.edu/LDC2016T23
2 https://archive.ics.uci.edu/ml/datasets/Twenty+Newsgroups
3 http://thuctc.thunlp.org
5 https://archive.ics.uci.edu/ml/datasets/WikiEvents
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 illustrates 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.

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 \(e = \langle t, a \rangle\), 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.
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 $e = \langle t, a \rangle$, 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).

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

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, 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 minibatches and learning rates. Following (Kowsari et al., 2019), we use Precision(P), Recall(R) and F1 score as the evaluation metrics. We report the

9https://huggingface.co/models
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 \( O = \{start_r, end_r\} \), the output of Q&A Methods 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 \( O = \{start_{r_1}, end_{r_1}, start_{r_2}, end_{r_2}, \ldots, \} \). 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-
From 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 remains 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.

### 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 remains 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.

---

2. https://github.com/xinyadu/doc_event_role

---

<table>
<thead>
<tr>
<th>Methods</th>
<th>Normal Setting</th>
<th>Cross-domain Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EM</td>
<td>HM</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>R</td>
</tr>
<tr>
<td>BERT-Seq(sent)</td>
<td>68.3</td>
<td>24.7</td>
</tr>
<tr>
<td>BERT-Seq(chunk)</td>
<td>71.0</td>
<td>29.9</td>
</tr>
<tr>
<td>BERT-Seq(doc)</td>
<td>69.1</td>
<td>33.5</td>
</tr>
<tr>
<td>BERT-QA(chunk)</td>
<td>60.4</td>
<td>33.1</td>
</tr>
<tr>
<td>Ontology-QA(chunk)</td>
<td>69.6</td>
<td>30.9</td>
</tr>
<tr>
<td>BART-Gen(chunk)</td>
<td>55.7</td>
<td>34.2</td>
</tr>
<tr>
<td>Doc2EDAG(chunk)</td>
<td>68.5</td>
<td>30.3</td>
</tr>
<tr>
<td>MG-Reader(seq+chunk)</td>
<td>69.3</td>
<td>30.1</td>
</tr>
<tr>
<td>Human</td>
<td>87.8</td>
<td>84.2</td>
</tr>
</tbody>
</table>

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


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.


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


