Open-Vocabulary Argument Role Prediction for Event Extraction

Yizhu Jiao Sha Li Yiqing Xie Ming Zhong Heng Ji Jiawei Han
University of Illinois at Urbana-Champaign, IL, USA
{yizhuj2, shal2, xyiqing2, mingz5, hengji, hanj}@illinois.edu

Abstract

The argument role in event extraction refers to the relation between an event and an argument participating in it. Despite the great progress in event extraction, existing studies still depend on roles pre-defined by domain experts. These studies expose obvious weakness when extending to emerging event types or new domains without available roles. Therefore, more attention and effort needs to be devoted to automatically customizing argument roles. In this paper, we define this essential but under-explored task: **open-vocabulary argument role prediction**. The goal of this task is to infer a set of argument roles for a given event type. We propose a novel unsupervised framework, RolePred for this task. Specifically, we formulate the role prediction problem as an in-filling task and construct prompts for a pre-trained language model to generate candidate roles. By extracting and analyzing the candidate arguments, the event-specific roles are further merged and selected. To standardize the research of this task, we collect a new event extraction dataset from WikiPpedia including 142 customized argument roles with rich semantics. On this dataset, RolePred outperforms the existing methods by a large margin. Source code and dataset are available on our GitHub repository[1].

1 Introduction

Great progress has been made on event extraction in recent years, however, most of the existing studies still rely on hand-crafted ontologies (Grishman and Sundheim, 1996; Ji and Grishman, 2008; Lin et al., 2020; Du and Cardie, 2020b; Liu et al., 2020; Zhou et al., 2021; Li et al., 2021b). Event ontologies such as Propbank (Kingsbury and Palmer, 2003) and FrameNet (Baker et al., 1998) take years, even decades, to construct. At the center of such ontologies lie argument roles, which capture the relation between an event and an argument participating in it. For instance, the `Transport` event type has 5 roles: `Agent`, `Artifact`, `Vehicle`, `Origin` and `Destination`. These roles are typically specific to the event type and semantically meaningful role names can directly benefit argument extraction quality. While human-constructed ontologies suffice for closed-domain applications, it requires extra human effort to extend to emerging event types or new domains. To overcome such difficulty, some studies attempt to automatically induce argument roles for given event types (Huang et al., 2016; Yuan et al., 2018; Liu et al., 2019a). These methods usually define a glossary including possible role names with general semantics, such as `Time`, `Place`, and `Value`, and then pick a subset as argument roles. Since role names are restricted to a limited vocabulary, they do not reflect the uniqueness of event types, such as the `Magnitude` of an earthquake, or the `Host` of a ceremony. Hence, predicting role names from an open vocabulary is necessary for broad coverage of event semantics.

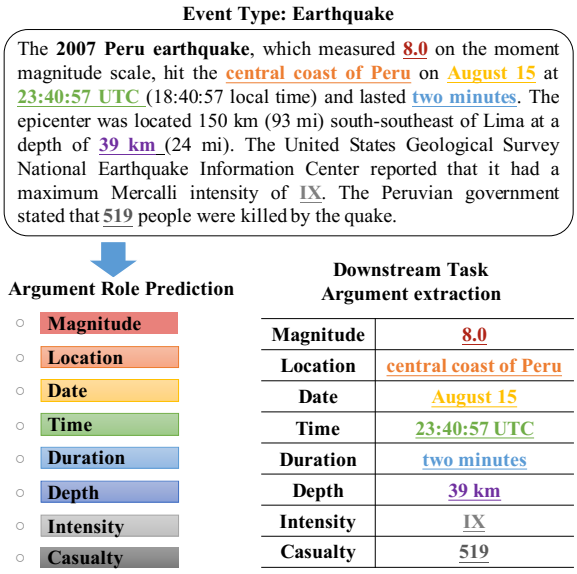


Figure 1: An example of the argument role prediction task and its downstream task.

[1] https://github.com/yzjiao/RolePred

Findings of the Association for Computational Linguistics: EMNLP 2022, pages 5404–5418
December 7-11, 2022 ©2022 Association for Computational Linguistics

In this paper, we introduce an essential but under-explored task for event extraction: **open-vocabulary argument role prediction**. This task aims to infer a set of argument role names for a given event type to describe the crucial relations between the event type and its arguments. As shown in Figure 1, for the `Earthquake` event type, given some related documents, we want to output key argument role names such as `magnitude`, `intensity`, `depth`, `deaths`, and `injuries`. These semantically meaningful roles can be directly used in the downstream event extraction task (Huang et al., 2018; Liu et al., 2020; Lyu et al., 2021). However, this task poses new challenges: (1) *decoupling argument role prediction from argument extraction*: For event extraction, roles and arguments are closely interdependent, one of which is pivotal to determining the other, and predicting argument roles for unknown arguments is a pressing problem; and (2) *customizing argument roles from an open vocabulary*: To cover board domains, we need to go beyond the predefined candidate vocabulary, and, the generated roles should be personalized for each event type so that they can reflect the unique features of different event types.

To tackle these challenges, we propose a novel unsupervised framework, ROLEPRED. Given an event type and a set of documents, ROLEPRED predicts the argument roles by three components including candidate role prediction, candidate argument extraction, and argument role selection. Concretely, to decouple roles from unknown arguments, we assume that named entities are more likely to be arguments. Based on this assumption, we regard the named entities in the text as possible arguments. Then, we predict their candidate role names by casting it as a prompt-based in-filling task (Raffel et al., 2020). Note that, we allow the pre-trained model (Raffel et al., 2020) to fill in a variable-length mask span instead of one single mask. Yet, those generated roles are still noisy. Therefore, considering the inter-dependency between roles and arguments, we extract arguments with QA models for further role selection and merging. Finally, the event-specific roles are obtained to serve for event extraction. In this way, generated roles are sufficiently fine-grained and event-specific.

Existing event extraction datasets have limited coverage of event types and insufficient refinement of argument roles (Grishman and Sundheim, 1996; Li et al., 2021b; Ebner et al., 2020). Thus, to support the research in argument role prediction, we collect a new event extraction dataset from Wikipedia named RoleEE. In statistics, our dataset contains 50 event types and 142 argument role types, much more than the number of argument roles in the existing dataset (5 in MUC-4 (Doddington et al., 2004) and 65 in RAMS (Ebner et al., 2020)). Besides the general roles, such as date and location, there are personalized roles for each event type, such as Accelerant for a Fire event, and Magnitude for an earthquake event, which carry rich semantics and assist to extract detailed arguments in events. Besides, our dataset focuses on the extraction of the main event in each document, that is, one-event-per-document. This setting discards the limitation that the event arguments exist within several consecutive sentences. Arguments scattering throughout the long document would be in line with real-world applications and present more challenges for an event extraction model. We set a baseline performance using ROLEPRED on this dataset and provide insights for future work.

2 Related work

Event Ontology Construction Event ontologies are a crucial prerequisite to event discovery and extraction. Great efforts have been paid in previous studies to build several high-quality ontologies, such as FrameNet (Baker et al., 1998), Propbank (Kingsbury and Palmer, 2003), and VerbNet (Kipper et al., 2008). However, it is costly and time-consuming to build hand-crafted ontologies. Some researchers start to explore automatic ontology construction. Specifically, much progress has been made in event schema induction to characterize the relationship among different events (Cheung et al., 2013; Peng and Roth, 2016; Li et al., 2020; Kwon et al., 2020; Li et al., 2021a). Also, several recent studies attempt to discover new event types from raw texts (Shen et al., 2021; Edwards and Ji, 2022). Nevertheless, as the center of event ontologies, argument role prediction has always been an underexplored task. Related studies (Yuan et al., 2018; Liu et al., 2019a) restrict role names to a limited vocabulary so that they fail to reflect the unique characteristics of different event types. Therefore, in this paper, we study an essential but challenging task: open-vocabulary argument role prediction.

Event Extraction This task has been mainly studied under two paradigms: (1) Sentence-level

Entity Type	Prompt
PERSON	*According to this, Entity play the role of ⟨MASK SPAN⟩in this Event Type.*
LOCATION	*According to this, the ⟨MASK SPAN⟩is Entity in this Event Type.*
NUMBER	*According to this, the number of ⟨MASK SPAN⟩of this Event Type is Entity.*
OTHER TYPES	*According to this, the ⟨MASK SPAN⟩of this Event Type is Entity.*

Table 1: Prompt design for different types of entities.

event extraction (Doddington et al., 2004) has been studied since an early stage (Chen et al., 2015; Nguyen et al., 2016; Yang et al., 2018), with a few models gone beyond individual sentences to make decisions (Ji and Grishman, 2008; Liao and Grishman, 2010; Zhao et al., 2018); and (2) document-level event extraction has gained a lot of research attention recently (Sundheim, 1992; Du and Cardie, 2020a; Huang and Jia, 2021; Ma et al., 2022; Yang et al., 2021). This study further explores extracting arguments scattered throughout documents.

3 Method

RolePred contains three core components: candidate role generation, candidate argument extraction, and argument role selection (in Figure 2). The following formulates the task of argument role prediction and then describes each component in turn.

3.1 Task Formulation

Formally, given an event type and a set of documents $\mathcal{D}$, each document $d \in \mathcal{D}$ mainly describes one event instance e of the same type. The task of argument role prediction aims to predict a set of event-specific roles $\mathcal{R}$. Each role $r \in \mathcal{R}$ is a phrase or a cluster of phrases with similar semantics.

3.2 Candidate Role Generation

Entities are often actors or participants in events. Thus, in the absence of available arguments, we introduce named entities to generate some candidates for argument roles. Specifically, given an event type, for each document d, we use the off-the-shelf named entity recognition tool (Honnibal and Montani, 2017) to identify all entities, $\mathcal{A}$, from the text. Then, we treat these entities as possible arguments, and try to predict their roles. This process of candidate role generation is formulated as a mask-filling task. For each entity $a \in \mathcal{A}$, we construct a prompt with masked words to feed into the pre-trained language model. As a result, the model infers these masks as the role name of this entity by decoding its inner semantic knowledge. Such a prompt is constructed as follows:

Context. According to this, the ⟨MASK SPAN⟩of this Event Type is Entity.

Here *Context* refers to the paragraph which mentions the entity from the source document. It provides a detailed background description of the event and the entity. Note that to avoid misleading information, the irrelevant sentences after the entity are removed. Then, it is followed by a natural language template containing ⟨*Entity*⟩ and ⟨*Event Type*⟩ placeholders. During inference, these placeholders are replaced by the concrete event type and entity. ⟨*MASK SPAN*⟩ represents a span of masks whose length is variable. For example, given the event type of earthquake and the entity of 5:36 PM, the constructed prompt is as follows:

The 1964 Alaskan earthquake, also known as the Great Alaskan earthquake, occurred at 5:36 PM AKST on Good Friday, March 27. According to this, the ⟨MASK SPAN⟩of this earthquake is 5:36 PM.

In this case, ⟨*MASK SPAN*⟩ is expected to be filled with *time*, or *start time* as the argument role. Besides, considering the entity's general semantic type: a person, location, number, or other, we slightly alter the prompt construction to fluently and naturally support the procedure of unmasking argument roles. Details are listed in Table 1.

The constructed prompt is input into the encoder-decoder language model T5 (Raffel et al., 2020) for candidate role generation. The generation process models the conditional probability of selecting a new token given the previous tokens and the input to the encoder. Note that the length of ⟨*MASK SPAN*⟩ is not fixed for model filling. Inspired by SpanBERT (Joshi et al., 2020), T5 samples the number of text spans from a Poisson distribution (λ = 3). Each span is replaced with a single token. By infilling the marked text, the model is taught to predict how many tokens are missing from a span. Therefore, our roles generated by the language model are customized phrases of various lengths according to the semantics of constructed prompts Unlike existing work that uses single general words as role names (Huang et al., 2016; Yuan et al., 2018; Liu et al., 2019a), our roles are more fine-grained

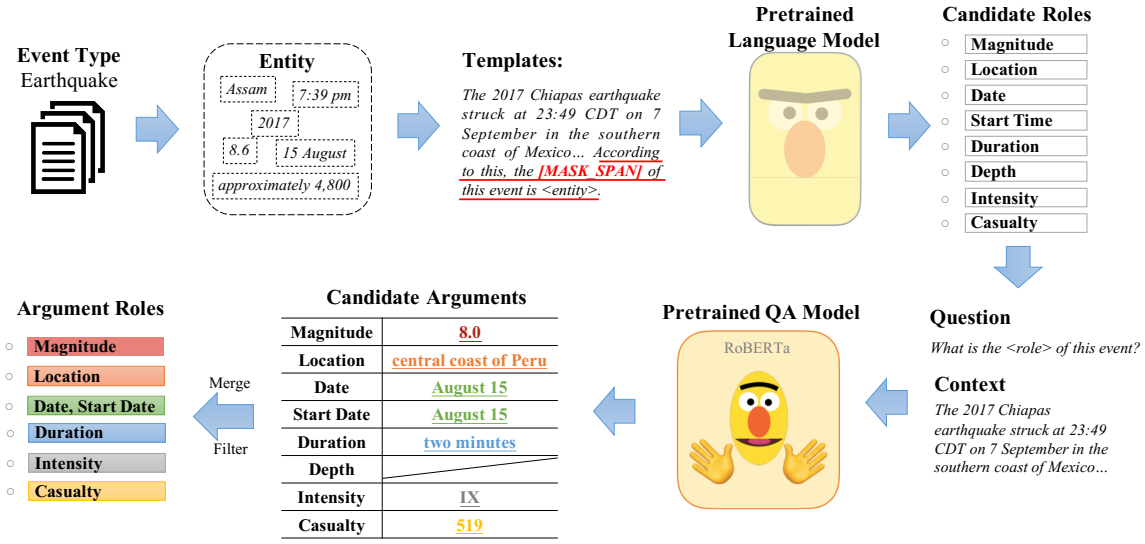


Figure 2: The framework of RolePred. It predicts argument roles by three components: first predict candidate role names for named entities by casting this problem as a prompt-based in-filling task, then extract candidate arguments for each candidate roles, and finally select the event-specific roles to serve for event extraction.

and contain more semantic details. This supports the subsequent task, argument extraction, to extract more participants for events from texts. Finally, the language model generates 10 possible argument roles per entity. For each document, we integrate the candidate role names of all entities for further selection.

3.3 Candidate Argument Extraction

For an event type, its salient argument roles are usually shared by most event instances. For example, each earthquake event has a magnitude but does not necessarily cause tsunamis. Therefore, it leaves the challenge of identifying relevant and salient roles for the candidates. Intuitively, arguments provide a feasible solution considering their strong interdependence with event roles. Along these lines, we first extract candidate arguments from each document for all candidate roles, and then conduct role selection using these arguments (more details in the next section).

Inspired by some existing work on argument extraction (Lyu et al., 2021), we formulate this problem into a question-answering task. Given an event type and a candidate role, we construct a question, which is fed into a standard pre-trained bidirectional transformer (BERT Devlin et al. (2018), RoBERTa Liu et al. (2019b)) along with a document. The QA model serves to identify candidate event arguments (spans of text) from each source document. Regarding the input sequences, we follow a standard BERT-style format as follows:

[CLS] What is the Event Role in this Event Type event? [SEP] Document [SEP]

Here, *[CLS]* is BERT's special classification token, *[SEP]* is the special token to denote separation, and *Document* is the tokenized input document. For example, given the event type of pandemic, the event role of casualty, and a document on COVID-19, the input sequences are as follows:

[CLS] What is the casualty in this pandemic event? [SEP] The COVID-19 pandemic is an ongoing global pandemic of coronavirus disease. It's estimated that the worldwide total number of deaths has exceeded five million ... [SEP]

In this case, the argument is expected to be *five million*. Note that, for some roles, a given document may not mention its argument. That is, the above-constructed question can be unanswerable. Thus, for each extracted answer, we set a threshold on its probability from the QA model to filter out some unreliable results. Besides, because our dataset focuses on one main event per document, unlike related work on sentence-level event extraction (Huang and Ji, 2020; Liu et al., 2020; Ma et al., 2022), we need to search for arguments throughout the document. This task is more challenging and well worth further exploration.

So far, in every document, for each candidate role, one candidate argument has been extracted. Thus, these argument-role pairs can be composed into one event instance per document.

3.4 Argument Role Selection

After extracting the main event instance from each document, candidate roles are selected with mainly two steps: argument role filtering and merging. Specifically, for an event type, its different event instances may present different attributes. These instances, however, usually have several common and significant argument roles (e.g., the intensity of the earthquake events and the host for the award ceremony events). Thus, we judge the salience of an argument role by involving multiple event instances of the same type. It is assumed that a role name belongs to the event type only if most of the event instances have their associated argument.

Regarding argument role merging, different roles can represent similar semantics and share the same arguments in an event. For example, the date, official date, and original date usually refer to the same day for a firework event. By merging similar role names, we can increase their specificity while reducing their number, thereby improving the efficiency of the subsequent argument extraction step. Along this line, we determine the semantic similarity of two roles based on the frequency that they share the same argument in the event instances. For example, given 10 instances of the blizzard event, if two roles, data, and official date, have the same day as their arguments in 5 instances, then their similarity is 0.5. We set a threshold to select semantically similar argument roles and merge them.

4 Dataset Construction

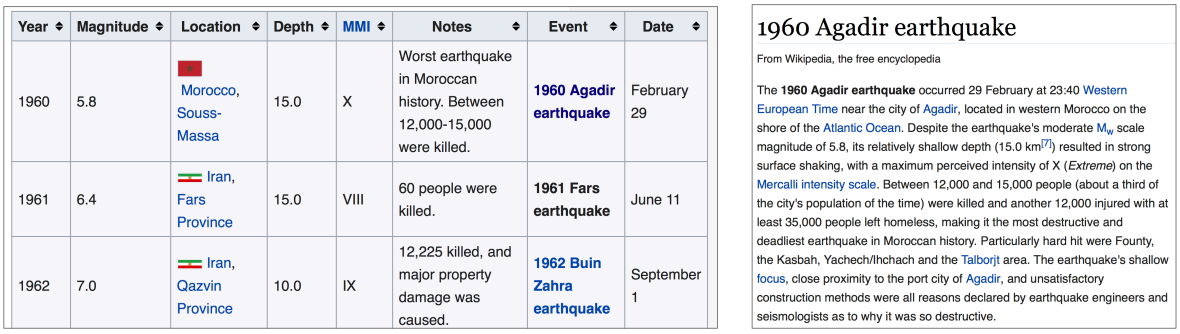
4.1 Data Collection

Event Type Selection. Among the hot topics in the journalism, we carefully select 50 impactful event types, such as earthquake, civil unrest and military occupation. To broaden the domain coverage, these event types cover many fields including politics, academia, art, sport, military, astronomy, and economics. Since these events usually contain rich argument roles, they require multiple sentences to describe. Thus, it is more suitable for document-level event extraction. More detailed examples are shown in Appendix A.

Argument Role Design. To construct the event-specific argument roles, we leverage the list of events in Wikipedia. Such a list shows the key attributes of multiple event instances of the same type. For example, Figure 3 show that the Wikipedia presents a list of recent major earthquakes. Their attributes can be regarded as the prototype arguments of the event type, such as `Year`, `Magnitude`, `Location` and `Depth`. Based on this observation, we search for one wiki list for each event type, and use the attributes as our basic set of argument roles. Then, we manually process these argument roles: (1) change abbreviations to common full names, such as `MMI` to `Magnitude`, (2) turn event names to triggers (`Name` or `Event` in the Wikipedia lists usually refer to the names of the event instances, which can be regarded as triggers), and (3) remove `Notes` which adds extra details to the event instances, but not suitable to be an argument role. With such manual annotation from Wikipedia, we design customized argument roles for each event type.

Event Argument Annotation. For each event type, the Wikipedia list usually involves multiple event instances. Each row in the list presents the information about one event. The values of each row can be regarded as the arguments of an event. For example, for the event "1960 Agadir earthquake", its magnitude is 5.8. Further cleaning is conducted on event instances to ensure quality: The event instances with incomplete arguments (e.g., null values or obvious errors) are dropped and the event instances whose source documents are inaccessible are removed (document acquisition is introduced in the next section). For the qualified events, their arguments are carefully refined by hand: (1) save only the arguments of the selected roles, (2) remove the special symbols or references in the arguments and keep only the key information, and (3) discard the arguments which are not mentioned in the corresponding documents (since they may come from other sources and cannot be extracted from our documents). Finally, for each event type, we obtain multiple event instances.

Source Document Acquisition. For each event instance, we adopt its Wikipedia article as the source document where the event arguments are annotated. Specifically, the Wikipedia lists usually mention the event name and provide the URL of the corresponding Wikipedia article. For example, as shown in Figure 3, the first earthquake event is linked to the Wikipedia article of 1960 Agadir earthquake. These documents describe one major event and usually mention most of the event arguments in the Wikipedia lists. Otherwise, those arguments will be cleaned up. We ensure that each event instance has a source document. Besides, the documents with less than 4 sentences are removed.



Year	Magnitude	Location	Depth	MMI	Notes	Event	Date
1960	5.8	Morocco, Souss-Massa	15.0	X	Worst earthquake in Moroccan history. Between 12,000-15,000 were killed.	**1960 Agadir earthquake**	February 29
1961	6.4	Iran, Fars Province	15.0	VIII	60 people were killed.	**1961 Fars earthquake**	June 11
1962	7.0	Iran, Qazvin Province	10.0	IX	12,225 killed, and major property damage was caused.	**1962 Buin Zahra earthquake**	September 1

(a) List of events on Wikipedia

1960 Agadir earthquake

From Wikipedia, the free encyclopedia

The **1960 Agadir earthquake** occurred 29 February at 23:40 Western European Time near the city of Agadir, located in western Morocco on the shore of the Atlantic Ocean. Despite the earthquake's moderate M_w scale magnitude of 5.8, its relatively shallow depth (15.0 km[7]) resulted in strong surface shaking, with a maximum perceived intensity of X (*Extreme*) on the Mercalli intensity scale. Between 12,000 and 15,000 people (about a third of the city's population of the time) were killed and another 12,000 injured with at least 35,000 people left homeless, making it the most destructive and deadliest earthquake in Moroccan history. Particularly hard hit were Founty, the Kasbah, Yachech/Ihchach and the Talborjt area. The earthquake's shallow focus, close proximity to the port city of Agadir, and unsatisfactory construction methods were all reasons declared by earthquake engineers and seismologists as to why it was so destructive.

(b) Source document

Figure 3: Data source of RoleEE. The left is a list of events from Wikipedia, from which we collect the argument roles and event instances on one event type. Each event instance has a URL pointing to its own Wiki page as shown in the right. We obtain the source documents from these Wiki page.

Datasets	# EvTyp.	# RoleTyp.	# Doc.	# ArgScat.
ACE2005	33	35	599	1
KBP2016	18	20	169	1
KBP2017	18	20	167	1
MUC-4	4	5	1,700	4.0
WikiEvents	50	59	246	2.2
RAMS	139	65	3,993	4.8
RoleEE	50	142	4,132	7.1

Table 2: Statistics of EE datasets. # EvTyp.: the number of event types. # RoleTyp.: the number of unique argument roles. # Doc.: the total number of documents. # ArgScat.: the number of sentences in which event arguments of the same event are scattered.

4.2 Data Analysis

In total, RoleEE contains 50 event types and 142 unique argument roles. Each event type has 5.2 argument roles on average. It labels 4,132 valid document-level events and 15,562 event arguments. The event type of Championship has the highest average number of event arguments per document (8.5). We compare RoleEE to various representative event extraction datasets in Table 2, including sentence-level EE datasets ACE2005 and KBP, and document-level EE dataset MUC-4, Wikievents, and RAMS. We find that RoleEE shows an advantage of rich argument role types, more than existing datasets. Besides some common argument roles, there are many unique role names customized for each event type. Thus, our dataset is more versatile in this aspect. In addition, RoleEE increases the difficulty in argument scattering, which is the critical challenge of document-level event extraction. We count the number of sentences in which event arguments of the same event are scattered. As shown in Table 2, the sentence-level EE event datasets only focus on one sentence, whereas our arguments are the most widely scattered among the document-level EE datasets, averaged with 7 sentences. It calls for subsequent work to pay more attention to this challenge.

5 Experiment

In this section, we first study the performance on the argument role prediction task, then examine the performance on the downstream task, argument extraction. Finally, we report our case analysis.

5.1 Argument Role Prediction

Implementation Details about our implementation are introduced in Appendix B.1.

Baselines Our method is compared with four existing baselines: LiberalEE (Huang et al., 2016), VASE (Yuan et al., 2018), ODEE (Liu et al., 2019a) and CLEVE (Wang et al., 2021) (More information in Appendix B.2). For ablation study, we evaluate three variants of RolePred: (1) - RoleMerge: it removes the similar role merging component from RolePred but still uses candidate arguments to filter those uncritical candidate roles; (2) - RoleMerge - RoleFilter: it removes two components from the full model, including similar role merging and unimportant role filtering, which are introduced in Section 2.4; and (3) RolePred (BERT): it adopts the same architecture of the full model while using the base version of BERT (Kenton and Toutanova, 2019) to generate candidate argument roles as introduced in Section 2.2. As to our full model, the base version of T5 (Raffel et al., 2020) is utilized for candidate role generation. In addition, we evaluate the human performance by inviting 3 PhD students who are not the authors of this paper to conduct this task manually. For each event

Models	Hard Matching			Soft Matching		
	Precision	Recall	F1	Precision	Recall	F1
LiberalEE	0.1342	0.2613	0.1773	0.3474	0.5340	0.4209
VASE	0.0926	0.1436	0.1125	0.2581	0.4274	0.3218
ODEE	0.1241	0.3076	0.1768	0.3204	0.4862	0.3862
CLEVE	0.1363	0.2716	0.1815	0.3599	0.5712	0.4415
RolePred (BERT)	0.2128	0.4582	0.2906	0.4188	0.6896	0.5211
RolePred (T5)	**0.2552**	**0.6461**	**0.3659**	**0.4591**	**0.7079**	**0.5570**
- RoleMerge	0.2233	0.6962	0.3381	0.4234	0.7677	0.5457
- RoleMerge - RoleFilter	0.1928	0.6582	0.2983	0.4188	0.7084	0.5264
Human	0.6098	0.8270	0.7020	0.7365	0.8732	0.7990

Table 3: Results of argument role prediction on our benchmark. Besides comparing with baselines, we also conduct the ablation study: the role merging and filtering are removed to verify their effectiveness.

type, each student is given the type name and 20 randomly selected documents. Then, each assessor writes down less than 20 argument roles, which are of less than three words. We average the scores of 3 students as the final human performance.

Evaluation Metrics Following previous studies (Liu et al., 2019a), we use precision, recall and F1-score as the metrics for argument role prediction. Two kinds of matching strategies are adopted: hard matching and soft matching. The former requires that the generated argument role and groundtruth should have at least one word in common; whereas the latter aims to compute the semantic similarity of a pair of roles. Specifically, given two roles, we use a pre-trained bidirectional transformer, Sentence-BERT (Reimers and Gurevych, 2019), to obtain their embeddings, and then calculate their cosine similarity as the semantic similarity score. Note that for multiple roles that are merged together, we concatenate them as one phrase for evaluation.

Experimental Result The evaluation results are shown in Table 3, with the following observations. (1) Compared with the existing methods, RolePred achieves significant improvements of 18.4% and 10.6% on F1 scores with hard and soft matching respectively. We speculate because argument roles are from open vocabulary. This speculation is verified by checking other baselines. LiberalEE and CLEVE perform relatively well since their roles come from hand-crafted knowledge bases. ODEE limits the roles to eight words, and the results validate its weaknesses. VASE can't get roles explicitly, thus failing in the comparison. (2) In the ablation study, even removing the merging and filtering parts, the variant of our method still outperforms the baselines, especially on hard matching. Based on this, role filtering provides a 4.0% and 1.9% improvement on F1 scores. The clear improvement of 2.7% and 1.2% occurs when we further merge similar roles. As a base model, T5 generates better roles than BERT. (3) The recall scores of RolePred can reach 64% and 70% on hard and soft matching respectively. This indicates that the generated argument roles can cover most of the groundtruth. Likely, it benefits from a lot of diverse roles which involve various aspects of event types. However, due to a large number of roles, the precision scores are reduced. It suggests we carefully select important and relevant roles to ensure the efficiency of event extraction.

Models	P	R	F1
LiberalEE	0.2009	0.2941	0.2387
VASE	0.2123	0.3257	0.2570
ODEE	0.2402	0.3712	0.2917
CLEVE	0.3529	0.3890	0.3701
RolePred (BERT)	0.4170	0.4333	0.4250
RolePred (Roberta)	**0.4131**	**0.5774**	**0.4817**
- RoleMerge	0.3855	0.6187	0.4750
- RoleMerge - RoleFilter	0.4397	0.5001	0.4679
RolePred (Gold Roles)	0.6664	0.4948	0.5679

Table 4: Results of argument extraction w/o gold roles. Besides the baselines, the argument merging and filtering are removed for ablation study.

5.2 Downstream Task

We conduct experiments to investigate the effect of argument role prediction on its downstream task: argument extraction. This task aims to identify arguments directly from raw texts without available roles of the given event type.

Evaluation Metrics We report precision (P), recall (R) and F1 scores as evaluation results. Note that the event arguments may have multiple men-

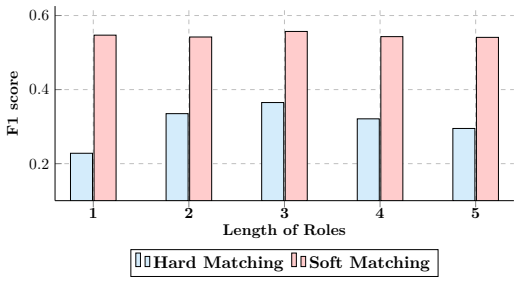


Figure 4: Impact of different length of role generation.

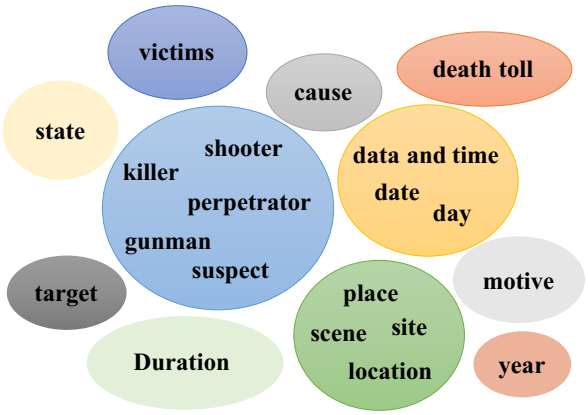


Figure 5: An example of our generated roles. The event type is `Shooting`. Each cluster has similar roles.

tions. For example, the location of a fire can be a country, state, or city. Therefore, we only require the extracted arguments to partially match with the groundtruth. In addition, for those arguments of the date or time type, we normalize[2] them into a uniform format for reasonable evaluation.

Baselines LiberalEE, VASE, ODEE and CLEVE are our baselines, the same as the setting of argument role prediction. Considering these methods extract multiple events from each document, we evaluate each with the groundtruth and then choose the highest score. Besides, we also study three variants of RolePred for ablation study:

For ablation study, we evaluate three variants of RolePred: (1) - RoleMerge: it removes the similar role merging component from RolePred but still uses candidate arguments to filter those uncritical candidate roles; (2) - RoleMerge - RoleFilter: it removes two components from the full model, including similar role merging and unimportant role filtering, which are introduced in Section 2.4; and (3) RolePred (BERT): it adopts the same architecture of the full model while using the base version of BERT (Kenton and Toutanova, 2019) to extract candidate arguments as introduced in Section 2.3. As to our full model, the base version of Roberta (Liu et al., 2019c) is utilized for candidate argument extraction. In addition, to explore the effect of role quality on downstream tasks, RolePred(Gold Roles) predict arguments using the true roles.

Experimental Result RolePred and all its variants outperform other baselines by a large margin, as shown in Table 4, This is likely because more specific role names can provide more semantics, thus assisting the model in identifying the correct arguments. When comparing RolePred and its variants, a similar trend is observed under the evaluation setting. By selecting salient roles, RolePred improves the effectiveness of argument extraction and increases the F1 score by 0.8%. Besides, by comparing with RolePred (Gold Roles), we can find that gold roles can greatly improve the precision score of argument extraction. Due to the large number of role names generated by the model, RolePred extracts more arguments and achieves a higher recall. Overall, given the high-quality roles, the f1 score of argument extraction is improved by 8%.

5.3 Impact of Role Length

To explore the effect of role length on our task, we set different maximum lengths for candidate role generation. Here we study the changing trend of f1 score using hard and soft matching. According to Figure 4, as the length increases from 1 to 5, the hard matching score shows a trend of increasing first and then decreasing, reaching a peak when the length is 3. This shows that long roles can be somewhat fine-grained, but too much detail will introduce noises. In addition, soft matching is not sensitive to this parameter. We speculate that because the short role already covers key elements.

5.4 Case Study

An example of our generated roles is displayed in Figure 5. The event type is `Shooting`. Here, the roles with similar semantics are merged into the same cluster, such as killer, shooter, and suspect. We can see that each cluster has various and salient roles for the shooting event. In addition, we also show a comparison of our model with baselines in Figure 6 for the argument extraction task (one representative role is picked from the clusters). Benefit from rich roles, our model is able to ef-

[2] https://dateutil.readthedocs.io/en/stable/parser.html

Output of RolePred

Victims	Maura Binkley and Nancy Van Vessem
State	Florida
Date	November 2, 2018
Killer	Scott Paul Beierle
Place	The yoga studio
Time	5:37 p.m. EDT
Duration	three and a half minutes
Motive	hatred of women
Target	Tallahassee Hot Yoga, a yoga studio
Year	2018

Output of ODEE

Agent	The gunman
Patient	six women

Output of CLEVE

Agent	Scott Paul Beierle
Patient	six women
Time	2018

Figure 6: An example of the extracted events by different methods including ODEE, CLEVE, and RolePred. The event type is `Shooting` and the event instance is 2018 Tallahassee Shooting.

fectively capture all arguments, while ODEE and CLEVE struggle with rare role types and result in uninformative extraction. More cases can be found in Appendix B.3.

5.5 Discussion on Data Leakage

Since our argument extractor relies on RoBERTa trained on SQuAD v2.0 dataset, which comes from the same source of our constructed dataset RoleEE, it might lead to the data leakage risk. Thus, we exclude articles used in SQuAD v2.0 from RoleEE when constructing the dataset. Specifically, we compare all articles in our dataset with SQuAD2.0, and count the number of articles that share sentences with SQuAD2.0. Here we only consider sentences of more than 4 words. As the result, we remove all the overlapping articles from RoleEE. In this paper, the dataset statistics and the experiment results are reported after this process.

6 Conclusion

This paper studies a challenging but essential task: open-vocabulary argument role prediction, and propose a novel unsupervised framework RolePred as a strong baseline and a carefully designed event extraction dataset for future work.

Limitations

RolePred is proposed based on the assumption that most arguments are named entities. It mainly focuses on entity arguments in raw texts. However, although non-entity arguments are relatively rare, they also play an important semantic part in lots of events. Our framework may get hindered when predicting roles for such non-entity arguments. Therefore, our next step is broader coverage of roles for different types of arguments.

In addition, our framework takes a set of related documents as input. It requires sufficient event instances for salient role selection. Also, the quality of generated argument roles heavily depends on document selection. Thus, for the given event type, retrieving representative documents of limited quantity can be considered an interesting topic for argument role prediction.

Furthermore, most of the existing work defines argument roles for an event type rather than an individual event instance. These argument roles are shared by multiple event instances of the same type. Nevertheless, different event instances can have personalized characteristics. For example, Magnitude is an argument role shared by all earthquakes, but Number of Landslides Caused can be a specific role to certain earthquakes. These specific roles can assist to identify specified and important arguments for event extraction. Accordingly, we expect to customize roles for one event instance in future work.

Acknowledgements

We thank Muhao Chen at USC, Yunyi Zhang, Tingcong Liu, and Yu Meng at UIUC for their helpful discussions and feedback. We would also like to thank anonymous reviewers for valuable comments and suggestions. Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004 and INCAS Program No. HR001121C0165, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897, and the Institute for Geospatial Understanding through an Integrative Discovery Environment (I-GUIDE) by NSF under Award No. 2118329. Any opinions, findings, and conclusions or recommendations expressed herein are those of the authors and do not necessarily represent the views, either expressed or implied, of DARPA or the U.S. Government. The views and conclusions contained in this paper are those of the authors and should not be interpreted as representing any funding agencies.

References

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 167–176.

Jackie Chi Kit Cheung, Hoifung Poon, and Lucy Vanderwende. 2013. Probabilistic frame induction. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 837–846.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Xinya Du and Claire Cardie. 2020a. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8010–8020.

Xinya Du and Claire Cardie. 2020b. Event extraction by answering (almost) natural questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 671–683.

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.

Carl Edwards and Heng Ji. 2022. Semi-supervised new event type induction and description via contrastive loss-enforced batch attention. *arXiv preprint arXiv:2202.05943*.

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

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings, convolutional neural networks and incremental parsing. *To appear*, 7(1):411–420.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare Voss, Jiawei Han, and Avirup Sil. 2016. Liberal event extraction and event schema induction. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 258–268.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2160–2170.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 340–351.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through unsupervised cross-document inference. In *In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL 2008). Ohio, USA*.

Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. 2020. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Paul Kingsbury and Martha Palmer. 2003. Propbank: the next level of treebank. In *Proceedings of Treebanks and lexical Theories*, volume 3. Citeseer.

Karin Kipper, Anna Korhonen, Neville Ryant, and Martha Palmer. 2008. A large-scale classification of english verbs. *Language Resources and Evaluation*, 42(1):21–40.

Heeyoung Kwon, Mahnaz Koupaee, Pratyush Singh, Gargi Sawhney, Anmol Shukla, Keerthi Kumar Kallur, Nathanael Chambers, and Niranjan Balasubramanian. 2020. Modeling preconditions in text with a crowd-sourced dataset. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3818–3828.

Manling Li, Sha Li, Zhenhailong Wang, Lifu Huang, Kyunghyun Cho, Heng Ji, Jiawei Han, and Clare Voss. 2021a. The future is not one-dimensional: Complex event schema induction by graph modeling for event prediction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5203–5215.

Manling Li, Qi Zeng, Ying Lin, Kyunghyun Cho, Heng Ji, Jonathan May, Nathanael Chambers, and Clare Voss. 2020. Connecting the dots: Event graph schema induction with path language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 684–695.

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

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 789–797.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651.

Xiao Liu, He-Yan Huang, and Yue Zhang. 2019a. Open domain event extraction using neural latent variable models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2860–2871.

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

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

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? paie: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6759–6774.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 300–309.

Haoruo Peng and Dan Roth. 2016. Two discourse driven language models for semantics. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 290–300.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Jiaming Shen, Yunyi Zhang, Heng Ji, and Jiawei Han. 2021. Corpus-based open-domain event type induction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5427–5440.

Beth M Sundheim. 1992. Overview of the fourth message understanding evaluation and conference. In *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. Cleve: contrastive pre-training for event extraction. *arXiv preprint arXiv:2105.14485*.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. Dcfee: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55.

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

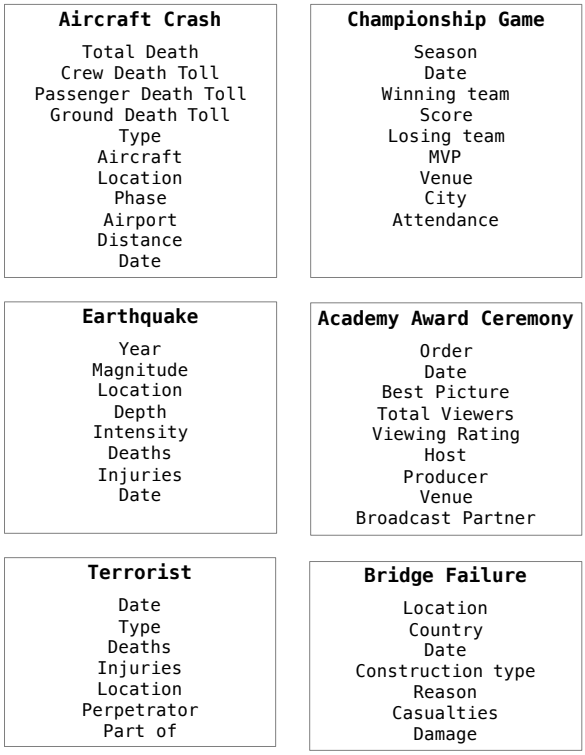


Figure 7: Examples of argument roles in our dataset.

Quan Yuan, Xiang Ren, Wenqi He, Chao Zhang, Xinhe Geng, Lifu Huang, Heng Ji, Chin-Yew Lin, and Jiawei Han. 2018. Open-schema event profiling for massive news corpora. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, pages 587–596.

Yue Zhao, Xiaolong Jin, Yuanzhuo Wang, and Xueqi Cheng. 2018. Document embedding enhanced event detection with hierarchical and supervised attention. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 414–419, Melbourne, Australia. Association for Computational Linguistics.

Yang Zhou, Yubo Chen, Jun Zhao, Yin Wu, Jiexin Xu, and Jinlong Li. 2021. What the role is vs. what plays the role: Semi-supervised event argument extraction via dual question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14638–14646.

A Dataset

In this section, we present more details about our dataset. All event types and the number of their corresponding documents are listed in Table 5. In addition, we show some examples of argument roles of our dataset in Figure 7. Also, more examples of event instances are in Figure 9.

B Experiment

B.1 Implementation

To identify named entities from raw texts, we use the off-the-shelf named entity recognition tool from the SpaCy library[3]. For candidate role generation, we adopt the base version of T5 (Raffel et al., 2020) as the pretrained generation model. The model is built based on the Huggingface[4]'s implementation with default parameters. The length of the constructed prompt is truncated to 512. For each prompt, the model generates 10 sequences whose maximum length is 3. The number of beams for beam search is set as 200. For candidate argument extraction, we use the large version of RoBERTa (Liu et al., 2019c) which has been trained on the SQuAD v2.0 benchmark (Rajpurkar et al., 2016). Its hyperparameters also refer to the Huggingface's implementation. For the extracted argument, if its probability from the model is below 0.3, the argument is discarded. For argument role filtering, given a role, when less than 40% of the documents mention its corresponding argument, it will be filtered out. For argument role merging, given a pair of roles, if they share the same argument in more than 50% of the documents, they will be merged together. We use one V100 GPUs with 32G memory for model training and evaluation. The prediction procedure lasts for about one day. For all the experiments, we report the average result of five runs as the final result. We also randomly select 20 documents for each event type and invite three students to annotate them for human evaluation.

B.2 Baselines

(1) LiberalEE (Huang et al., 2016): it leverages Abstract Meaning Representation to represent event structures and its argument roles are mapped with role descriptions in existing event knowledge bases (Baker et al., 1998; Kingsbury and Palmer, 2003); (2) VASE (Yuan et al., 2018): it proposes a Bayesian non-parametric model to obtain event profiles and represents argument roles with a tuple of entity roles; (3) ODEE (Liu et al., 2019a): it constructs a latent variable neural model to extract unconstraint types of events from news clusters. and chooses argument roles from 8 possible reference words; and (4) CLEVE (Wang et al., 2021): it provides a contrastive pre-training framework to

[3] https://spacy.io/
[4] https://huggingface.co/models

learn event knowledge and follows the pipeline of LiberalEE to discover argument roles.

B.3 Case Study

To study each component in our framework, we show two examples of their outputs given two event types of Earthquake and Pandemic in Figure 10. The example includes the generated candidate roles, the extracted candidate arguments, and the clusters of roles as the final model output. Here, the generated candidate roles are sorted by their importance scores. The extracted candidate arguments are from a randomly selected document. And the clusters of roles are ranked by the cluster size and the importance scores. In addition, Figure 8 presents four more extracted event instances. We remove the roles that have no available argument in the source document. From these cases, we can see that our method can actually extract informative and reasonable events with specific argument roles.

Event type: Blizzard

Trigger	Blizzard of 1977
Date	January 28 to February 1
Last day	February 1
Official name	blizzard of 1977
Year	1977
Onset	January 28
Average height	30 to 40 ft
Location	Western New York and Southern Ontario
Worst part	frequent whiteouts and zero visibility

Event type: LGBT Event

Trigger	Berlin Pride
Original name	Christopher Street Day Berlin
Organizer	Berliner CSD e.V
Main location	Berlin, Germany
First anniversary	June 30, 1979
Start date	usually starting at the end of May
Duration	month-long
Participants	lesbian, gay, bisexual, and transgender people and their allies

Event type: Solar Eclipse

Trigger	Solar eclipse of February 17, 2064
Date	February 17, 2064
Duration	12 minutes and 9 seconds
Primary object	Moon
Year	2064
Cause	Moon's apparent diameter is smaller than the Sun's
Frequency	every 358 synodic months
Time period	18 years, 11 days, and 8 hours
Symbol	annulus

Event type: Music Award Ceremony

Trigger	55th Academy of Country Music Awards
Host	Keith Urban
Venue	Nashville, Tennessee
Winner	Miranda Lambert
Official title	The 55th Academy of Country Music Awards
Host state	Tennessee
Date	September 16, 2020
Total number	55th

Figure 8: Four examples of event instances extracted by our framework.

Trigger	1988 Lancang–Gengma earthquakes
Year	1988
Magnitude	7.7
Location	Myanmar-China border region
Intensity	X
Deaths	938
Injuries	7,700
Date	November 6

Trigger	Kidnapping Kaede Ariyama
Date	17 November 2004
Victim Name	Kaede Ariyama
Abductor	Kaoru Kobayashi
Location	Nara, Japan
Age of victim	7
Outcome	Murdered

Trigger	The Garth Brooks World Tour with Trisha Yearwood
Artist	Garth Brooks and Trisha Yearwood
Year	1996
Number of Shows	366
Attendance	4 million
Actual gross	$364 million

Trigger	First inauguration of George Washington
Date	April 30, 1789
Location	Front balcony, Federal Hall New York, New York
Administer oath	Robert Livingston, Chancellor of New York
Address length	1431 words

Figure 9: Examples of event arguments in our dataset.

Event Type	**# Docs**	**Event Type**	**# Docs**	**Event Type**	**# Docs**
film festival	532	aviation accident	459	aircraft crash	390
massacre	222	kidnapping	216	explosion	190
flood	178	war	147	LGBT event	130
satellite launch	130	military occupation	117	bridge failure	100
shipwreck	94	disaster	91	sentence	91
human stampede	84	NBA final	77	concern tour	71
dam failure	68	inauguration	63	Olympic game	62
earthquake	61	tornado	57	railway terrorist	49
resignation	47	strike	47	academy award ceremony	44
avalanche	43	boiler explosion	39	blizzard	33
terrorist	32	firework	32	bank failure	32
civil unrest	29	extinction	28	music award ceremony	28
wildfire	26	bushfire	26	bankruptcy	23
protest	23	surfing competition	23	shooting	22
pandemic	21	rail accident	21	volcano eruption	21
recession	19	solar eclipse	17	nightclub fire	17
festival	17	championship game	14		

Table 5: All the event types and the numbers of corresponding documents in our dataset.

Event Type: Earthquake	**Event Type**: Pandemic
Candidate Roles official date epicenter date main cause main source magnitude location origin maximum magnitude survivors year Magnitudes main site average magnitude Depth geographical location fatalities victims total magnitude time main location source primary cause exact date cause site local time total number original date primary source casualties official name	**Candidate Roles** date cause origin epicenter death toll deaths most affected region total population source main cause victims name earliest record main target capital location most affected country largest affected region worst affected region global mortality rate duration fatality rate most vulnerable region outbreak cases peak earliest date year official start date most likely source originator confirmed cases
Candidate Arguments 'official date': ('July 30 at 05:11 UTC') 'estimated time': ('8.0') 'origin': ('moderate tsunami') 'actual date': ('July 30 at 05:11 UTC') 'official year': ('1995') 'official name': ('Antofagasta earthquake') 'local time': ('05:11 UTC') 'original year': ('1995') 'year': ('1995') 'original date': ('July 30 at 05:11 UTC') 'official language': ('Antofagasta') 'exact date': ('July 30 at 05:11 UTC') 'cause': ('moderate tsunami') 'date': ('July 30') 'official time': ('July 30 at 05:11 UTC') 'time': ('July 30 at 05:11 UTC') 'record': ('8.0 and a maximum Mercalli intensity of VII') 'current time': ('8.0') 'original name': ('Antofagasta earthquake') 'name': ('Antofagasta earthquake') 'main source': ('tsunami') 'main location': ('Antofagasta Region') 'epicenter': ('Antofagasta') 'fatalities': ('three')	**Candidate Arguments** 'originator': (H1N1 influenza A virus) 'primary source': (H1N1 influenza A virus) 'name': (The 1918 influenza pandemic) 'earliest date': (March 1918) 'earliest record': (The earliest documented case was March 1918 in Kansas, United States) 'main source': (H1N1 influenza A virus) 'beginning': (The earliest documented case was March 1918 in Kansas, United States) 'total population': (nearly a third of the global population, or an estimated 500 million people) 'year': (1918) 'death toll': (17 million to 50 million) 'source': (H1N1 influenza A virus) 'deaths': (17 million to 50 million) 'outbreak': (H1N1 influenza A virus) 'people who died': (17 million to 50 million) 'most likely cause': (H1N1 influenza A virus) 'most common name': (Spanish flu or as the Great Influenza epidemic) 'most common cause': (H1N1 influenza A virus) 'main cause': (H1N1 influenza A virus) 'birthplace': (Kansas, United States)
Model Output ['initial magnitude', 'average magnitude', 'magnitude', 'estimated magnitude', 'total magnitude', 'magnitudes', 'current magnitude'] ['primary cause', 'main cause', 'cause', 'major cause'] ['actual date', 'current date', 'exact date', 'date', 'official date', 'original date'] ['maximum intensity', 'intensity', 'maximum magnitude'] ['official year', 'original year', 'year'] ['location', 'main location'] ['duration', 'total duration'] ['fatalities', 'deaths'] ['total depth', 'depth'] ['epicenter'] ['average age'] ['maximum depth'] ['site'] ['confirmed victims'] ['current time'] ['total population']	**Model Output** [date, year, beginning date, official date, start date, exact date, original date] [geographical origin, geographic origin] [main cause, cause, primary cause] [period, time frame, time period] [name, official name] [deaths, fatalities] [total duration] [peak period] [main source] [peak season] [geographical center] [end date] [global population] [originator] [peak year] [primary target] [peak]

Figure 10: Example outputs of each component in our framework.