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 WikiPedia including 143 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 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

---

1https://github.com/yzjiao/RolePred
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 143 argument role types, much more than the number of argument roles in the existing dataset (5 in MUC-4 (Dodgington 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
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:

**Table 1: Prompt design for different types of entities.**

<table>
<thead>
<tr>
<th>Entity Type</th>
<th>Prompt</th>
</tr>
</thead>
<tbody>
<tr>
<td>PERSON</td>
<td>According to this, Entity play the role of (MASK SPAN) in this Event Type.</td>
</tr>
<tr>
<td>LOCATION</td>
<td>According to this, the (MASK SPAN)is Entity in this Event Type.</td>
</tr>
<tr>
<td>NUMBER</td>
<td>According to this, the number of (MASK SPAN)of this Event Type is Entity.</td>
</tr>
<tr>
<td>OTHER TYPES</td>
<td>According to this, the (MASK SPAN)of this Event Type is Entity.</td>
</tr>
</tbody>
</table>

Table 1: Prompt design for different types of entities. **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 $\langle$Entity$\rangle$ and $\langle$Event Type$\rangle$ placeholders. During inference, these placeholders are replaced by the concrete event type and entity. $\langle$MASK SPAN$\rangle$ 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 $\langle$MASK SPAN$\rangle$ of this earthquake is $\langle$time$\rangle$.***

In this case, $\langle$MASK SPAN$\rangle$ 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 $\langle$MASK SPAN$\rangle$ is not fixed for model filling. Inspired by SpanBERT (Joshi et al., 2020), T5 samples the number of text spans from a Poisson distribution ($\lambda = 3$). 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.
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.
List of events on Wikipedia

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.

<table>
<thead>
<tr>
<th>Datasets</th>
<th># EvTyp.</th>
<th># RoleTyp.</th>
<th># Doc.</th>
<th># ArgScat.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACE2005</td>
<td>33</td>
<td>35</td>
<td>599</td>
<td>1</td>
</tr>
<tr>
<td>KBP2016</td>
<td>18</td>
<td>20</td>
<td>169</td>
<td>1</td>
</tr>
<tr>
<td>KBP2017</td>
<td>18</td>
<td>20</td>
<td>167</td>
<td>1</td>
</tr>
<tr>
<td>MUC-4</td>
<td>4</td>
<td>5</td>
<td>1,700</td>
<td>4.0</td>
</tr>
<tr>
<td>WikiEvents</td>
<td>50</td>
<td>59</td>
<td>246</td>
<td>2.2</td>
</tr>
<tr>
<td>RAMS</td>
<td>139</td>
<td>65</td>
<td>3,993</td>
<td>4.8</td>
</tr>
<tr>
<td>RoleEE</td>
<td>50</td>
<td>143</td>
<td>4,132</td>
<td>7.1</td>
</tr>
</tbody>
</table>

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 143 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...
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.

<table>
<thead>
<tr>
<th>Models</th>
<th>Hard Matching</th>
<th>Soft Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>LiberalEE</td>
<td>0.1342</td>
<td>0.2613</td>
</tr>
<tr>
<td>VASE</td>
<td>0.0926</td>
<td>0.1436</td>
</tr>
<tr>
<td>ODEE</td>
<td>0.1241</td>
<td>0.3076</td>
</tr>
<tr>
<td>CLEVE</td>
<td>0.1363</td>
<td>0.2716</td>
</tr>
<tr>
<td>ROLEPRED (BERT)</td>
<td>0.2128</td>
<td>0.4582</td>
</tr>
<tr>
<td>ROLEPRED (T5)</td>
<td>0.2552</td>
<td>0.6461</td>
</tr>
<tr>
<td>- RoleMerge</td>
<td>0.2233</td>
<td>0.6962</td>
</tr>
<tr>
<td>- RoleMerge - RoleFilter</td>
<td>0.1928</td>
<td>0.6582</td>
</tr>
<tr>
<td>Human</td>
<td>0.6098</td>
<td>0.8270</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Models</th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>LiberalEE</td>
<td>0.2009</td>
<td>0.2941</td>
<td>0.2387</td>
</tr>
<tr>
<td>VASE</td>
<td>0.2123</td>
<td>0.3257</td>
<td>0.2570</td>
</tr>
<tr>
<td>ODEE</td>
<td>0.2402</td>
<td>0.3712</td>
<td>0.2917</td>
</tr>
<tr>
<td>CLEVE</td>
<td>0.3529</td>
<td>0.3890</td>
<td>0.3701</td>
</tr>
<tr>
<td>ROLEPRED (BERT)</td>
<td>0.4170</td>
<td>0.4333</td>
<td>0.4250</td>
</tr>
<tr>
<td>ROLEPRED (Roberta)</td>
<td>0.4131</td>
<td>0.5774</td>
<td>0.4817</td>
</tr>
<tr>
<td>- RoleMerge</td>
<td>0.3855</td>
<td>0.6187</td>
<td>0.4750</td>
</tr>
<tr>
<td>- RoleMerge - RoleFilter</td>
<td>0.4397</td>
<td>0.5001</td>
<td>0.4679</td>
</tr>
<tr>
<td>ROLEPRED (Gold Roles)</td>
<td>0.6664</td>
<td>0.4948</td>
<td>0.5679</td>
</tr>
</tbody>
</table>

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.

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-
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. Each cluster has similar roles. 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-
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

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.

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


Manling Li, Sha Li, Zhenhai-long 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 5195–5215.


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.


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. For candidate role generation, we adopt the base version of T5 as the pretrained generation model. The model is built based on the Huggingface’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 which has been trained on the SQuAD v2.0 benchmark. 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

3https://spacy.io/
4https://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.

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Event type: Blizzard</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>January 28 to February 1</td>
</tr>
<tr>
<td>Last day</td>
<td>February 1</td>
</tr>
<tr>
<td>Official name</td>
<td>blizzard of 1977</td>
</tr>
<tr>
<td>Year</td>
<td>1977</td>
</tr>
<tr>
<td>Onset</td>
<td>January 28</td>
</tr>
<tr>
<td>Average height</td>
<td>30 to 40 ft</td>
</tr>
<tr>
<td>Location</td>
<td>Western New York and Southern Ontario</td>
</tr>
<tr>
<td>Worst part</td>
<td>frequent whiteouts and zero visibility</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Event type: LGBT Event</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original name</td>
<td>Christopher Street Day Berlin</td>
</tr>
<tr>
<td>Organizer</td>
<td>Berliner CSD e.V</td>
</tr>
<tr>
<td>Main location</td>
<td>Berlin, Germany</td>
</tr>
<tr>
<td>First anniversary</td>
<td>June 30, 1979</td>
</tr>
<tr>
<td>Start date</td>
<td>usually starting at the end of May</td>
</tr>
<tr>
<td>Duration</td>
<td>month-long</td>
</tr>
<tr>
<td>Participants</td>
<td>lesbian, gay, bisexual, and transgender people and their allies</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Event type: Solar Eclipse</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date</td>
<td>February 17, 2064</td>
</tr>
<tr>
<td>Duration</td>
<td>12 minutes and 9 seconds</td>
</tr>
<tr>
<td>Primary object</td>
<td>Moon</td>
</tr>
<tr>
<td>Year</td>
<td>2064</td>
</tr>
<tr>
<td>Cause</td>
<td>Moon’s apparent diameter is smaller than the Sun’s</td>
</tr>
<tr>
<td>Frequency</td>
<td>every 358 synodic months</td>
</tr>
<tr>
<td>Time period</td>
<td>18 years, 11 days, and 8 hours</td>
</tr>
<tr>
<td>Symbol</td>
<td>annulus</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trigger</th>
<th>Event type: Music Award Ceremony</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host</td>
<td>Keith Urban</td>
</tr>
<tr>
<td>Venue</td>
<td>Nashville, Tennessee</td>
</tr>
<tr>
<td>Winner</td>
<td>Miranda Lambert</td>
</tr>
<tr>
<td>Official title</td>
<td>The 55th Academy of Country Music Awards</td>
</tr>
<tr>
<td>Host state</td>
<td>Tennessee</td>
</tr>
<tr>
<td>Date</td>
<td>September 16, 2020</td>
</tr>
<tr>
<td>Total number</td>
<td>55th</td>
</tr>
</tbody>
</table>

Figure 8: Four examples of event instances extracted by our framework.
Table 5: All the event types and the numbers of corresponding documents in our dataset.
<table>
<thead>
<tr>
<th>Event Type: Earthquake</th>
<th>Event Type: Pandemic</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Candidate Roles</strong></td>
<td><strong>Candidate Roles</strong></td>
</tr>
<tr>
<td>official date</td>
<td>date</td>
</tr>
<tr>
<td>epicenter</td>
<td>cause</td>
</tr>
<tr>
<td>date</td>
<td>origin</td>
</tr>
<tr>
<td>main cause</td>
<td>epicenter</td>
</tr>
<tr>
<td>main source</td>
<td>death toll</td>
</tr>
<tr>
<td>magnitude</td>
<td>deaths</td>
</tr>
<tr>
<td>location</td>
<td>most affected region</td>
</tr>
<tr>
<td>origin</td>
<td>total population</td>
</tr>
<tr>
<td>maximum magnitude</td>
<td>originator</td>
</tr>
<tr>
<td>survivors</td>
<td>source</td>
</tr>
<tr>
<td>year</td>
<td>cause</td>
</tr>
<tr>
<td>Magnitudes</td>
<td>main cause</td>
</tr>
<tr>
<td>main site</td>
<td>victims</td>
</tr>
<tr>
<td>average magnitude</td>
<td>victims</td>
</tr>
<tr>
<td>Depth</td>
<td>total number</td>
</tr>
<tr>
<td>geographical location</td>
<td>fatalities</td>
</tr>
<tr>
<td></td>
<td>fatalities</td>
</tr>
</tbody>
</table>

**Candidate Arguments**

| official date: 'official date': ('July 30 at 05:11 UTC') |
| estimated time: 'estimated time': ('8.0')               |
| origin: 'origin': ('moderate tsunami')                 |
| actual date: 'actual date': ('July 30 at 05:11 UTC')   |
| official year: 'official year': ('1995')               |
| local time: 'local time': ('05:11 UTC')                |
| original year: 'original year': ('1995')               |
| year: 'year': ('1995')                                 |
| date: 'date': ('July 30')                              |
| time: 'time': ('July 30 at 05:11 UTC')                 |
| record: 'record': ('8.0 and a maximum Mercalli intensity of VII') |
| current time: 'current time': ('8.0')                  |
| original name: 'original name': ('Antofagasta earthquake') |
| name: 'name': ('Antofagasta earthquake')               |
| main source: 'main source': ('tsunami')                |
| main location: 'main location': ('Antofagasta Region') |
| epicenter: 'epicenter': ('Antofagasta')                |
| fatalities: 'fatalities': ('three')                    |

**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'] |

| 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.