Prompt-Guided Few-Shot Event Detection

Anonymous ACL submission

Abstract

Practical applications of event extraction systems have long been hindered by their need for heavy human annotation. In order to scale up to new domains and event types, models must learn to cope with limited supervision, as in few-shot learning settings. To this end, the major challenge is to let the model master the semantic of event types, without requiring abundant event mention annotations. In our study, we employ cloze prompts to elicit event-related knowledge from pretrained language models and further use event definitions and keywords to pinpoint the trigger word. By formulating the event detection task as an *identify-then-localize* procedure, we minimize the number of type-specific parameters, enabling our model to quickly adapt to event detection tasks for new types. Experiments on three event detection benchmark datasets (ACE, FewEvent, MAVEN) show that our proposed method performs favorably under fully supervised settings and surpasses existing few-shot methods by 16% F1 on the FewEvent dataset and 23% on the MAVEN dataset when only 5 examples are provided for each event type. [1]

1 Introduction

Understanding events is central to information extraction, and event detection is an inevitable step in this process. The task of event detection is to locate the event trigger (i.e., the minimal lexical unit that indicates the event) and classify the trigger into one of the given event types. While steady progress has been made for event detection given ample supervision (Wadden et al., 2019; Lin et al., 2020; Lu et al., 2021), it is hard to replicate these success stories in new domains and on new event types without large-scale annotation. Here, to respond to emerging user needs and cope with limited annotation, we focus our study on the few-shot learning setting.

[1] Our model implementations and data preparation scripts will be made publicly available upon acceptance.

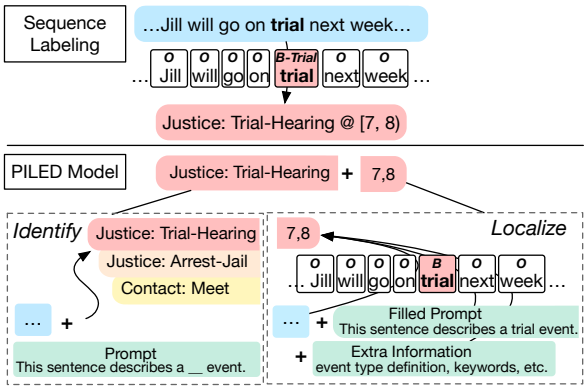

Figure 1: Event detection requires the model to produce both event types and trigger locations. Conventionally, it is formulated as a token-level sequence labeling problem. In our PILED (*Prompt-guided identify-then-localize event detection*) model, we decompose the task into two stages of identification and localization.

Recently, prompt-based learning has shown great success in few-shot learning for a range of classification and generation tasks. Compared to the typical supervised learning paradigm, prompt-based models are not only shaped by the annotated examples, but can also be guided by the prompt. Intuitively, in Figure 1, the prompt "The sentence describes a `[MASK]` event" makes the masked language model prediction resemble the event type mentioned in the context. Inspired by this approach, we tailor the cloze-based prompt learning paradigm for event detection.

Since event detection aims to recognize both the event type and the trigger location, the cloze-based prompt learning paradigm (Schick and Schütze, 2021a) is not directly applicable. In our study, we propose an *identify-then-localize* approach, which detaches the type semantic from the sequence labeling and opens the door to prompt learning. Specifically, we first recognize the event type (the *identification* stage) via a prompt-based multi-label model, and conduct trigger extraction based on the semantic type description (the *localization* stage).

Our identification model extends cloze-based prompt learning (Schick and Schütze, 2021a) to event detection. One key ingredient of prompt learning is the verbalizer: a mapping from the class label to a token in the language model's vocabulary. Since a sentence can contain multiple events, we extend the model to a multi-label classification setting by designating a special token as the verbalizer for the NULL class as well and comparing it against the predictions for all of the valid event types (as in Figure 2). In this design, the NULL verbalizer effectively serves as the dynamic threshold for multi-class classification (Zhou et al., 2021).

The localization model is a single-class sequence tagger that takes one identified event type as input and aims to recognize the corresponding trigger (as in Figure 3). Since we narrow the search to one event type, we employ the filled prompt, event descriptions, and keywords to augment the input. In this way, we decouple the model from the event label by including the event label information on the input side instead. This makes our localization model type-free, thus benefitting from the training examples of all event types.

We test our model on a range of datasets (ACE 2005, FewEvent (Deng et al., 2020), MAVEN (Wang et al., 2020)) under fully-supervised and few-shot event detection settings. Our experiments show that our model achieves state-of-the-art performance under the fully-supervised setting and dramatically outperforms existing baselines under the few-shot setting. Our main contributions include:

- We introduce a new *identify-then-localize* approach to event detection. By decoupling the type semantics from the sequence labeling task, we bring the benefits of cloze-based prompt learning to event detection and allow for flexible injection of event knowledge.
- We extend the cloze-based prompt learning paradigm to multi-label event type classification. This enables us to leverage the language modeling ability of pretrained LMs for the event identification task and adapt quickly to new event types. This method can apply to other multi-label classification tasks.
- We design an attention-enhanced single-class CRF tagger for event trigger localization. This attention mechanism allows for the interaction of predictions over neighboring tokens.
- Our model achieves excellent performance on the event detection task under both few-shot and fully-supervised settings. In particular, for few-shot event detection on FewEvent (Deng et al., 2020), we surpass the next best baseline by over 16% F1. On MAVEN, we achieve 8% F1 gains in the identification stage and present the first results for few-shot event detection.

2 Methodology

Given a collection of contexts $\mathcal{C}$ and a pre-defined event ontology $\mathcal{T}$ (a set of target event types), event detection aims to find all event mentions in the collection that fall into the given ontology. An event mention is characterized by a trigger span s (start index, end index) and an event type $t \in \mathcal{T}$. Here we follow previous work and consider each sentence as the context of the event.

We divide the event detection task into two stages: identification and localization. In the identification stage, for each context c, we find a set of event types T that have been mentioned. In the localization stage, we take a pair of context and event type (c, t) as input and find a set of spans S that correspond to the triggers for that event type. Note that both stages can produce a variable number of outputs for each input.

2.1 Event Type Identification

The event type identification model follows the idea of using a cloze-style prompt for few-shot learning with masked language models (Schick and Schütze, 2021a). Cloze-style prompt learning transforms a classification problem into a masked language modeling using a *prompt* and a *verbalizer* function. The *prompt* P is a natural language sentence with a `[MASK]` token. This prompt can be viewed as a cloze question, whereas the answer is related to the desired class label. Figure 2 shows a cloze prompt that can be used for event detection: "This text describes a `[MASK]` event".

The relationship between the class labels $\mathcal{L}$ and the predicted tokens V for the `[MASK]` is defined by the *verbalizer* function f_v: $\mathcal{L} \rightarrow V$. For example, we choose the verbalizer function to map the event type `Start-Position` to the token `hire`. We also refer to `hire` as the verbalizer for `Start-Position`.

During prediction, we use the logit that the masked language model M assigns to the verbalizer $f_v(l)$ for label l as the proxy for predicting l.

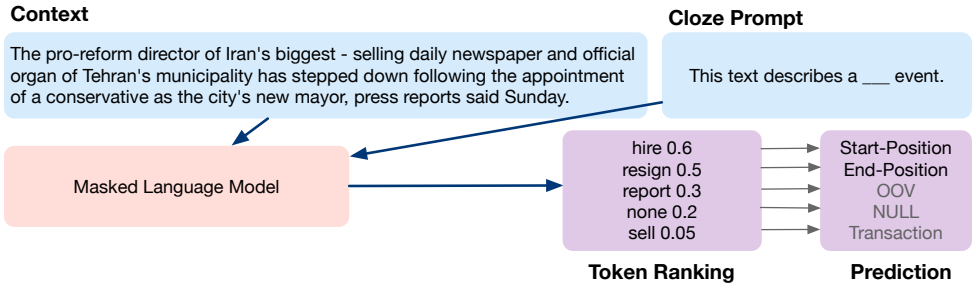

Figure 2: The identification model. The context and cloze prompt are concatenated and provided as input to the masked language model (MLM). The MLM produces scores for every token in the vocabulary as a measure of how well the token fits into the "blank". Some tokens in the vocabulary can be mapped back to event types, such as `hire` → `Start-Position`. If a token does not map to any event type in the ontology (e.g., `report`), it will be ignored. We predict all event types that have a higher score than the NULL label (which maps to the token `none`).

In the classification task, the probability for label l can then be computed as shown in Equation 1.

$$p(t = l) = \frac{\exp\left(M(f_v(l)|x, P)\right)}{\sum_{l' \in \mathcal{L}} \exp\left(M(f_v(l')|x, P)\right)} \quad (1)$$

For event detection, since each sentence can potentially mention multiple event types, we extend this approach to handle multi-label classification. Through the masked language model, we score all tokens in the vocabulary. After excluding tokens that do not map back to any event type of interest (such as the token `report` in the example), we obtain a ranking among all event types. The key becomes finding the cutoff threshold for translating these scores into outputs. We assign a token v_{NULL} to the NULL type[2] and use it as an adaptive threshold. In the inference stage, we predict all event types that score higher than the NULL type to be positive. In our example, since `hire` and `resign` both have higher scores than the NULL verbalizer `none`, we predict `Start-Position` and `End-Position` as the event types in the context.

During training, for each sentence, we compute the loss for the positive event types and the negative event types separately with respect to the NULL type:

$$\mathcal{L}_{\text{pos}} = \frac{1}{|T|} \sum_{t \in T} \log \frac{\exp(M(f_v(t)|x, P))}{\sum_{t' \in \text{NULL} \cup t} \exp(M(f_v(t')|x, P))} \quad (2)$$

where T is the set of positive event types for the sentence.

$$\mathcal{L}_{\text{neg}} = \log \frac{\exp(M(v_{\text{NULL}}|x, P)}{\sum_{t' \in \text{NULL} \cup \bar{T}} \exp(M(f_v(t')|x, P))} \quad (3)$$

[2]In our experiments, we use the token "none" as the NULL type's verbalizer.

$$\mathcal{L}_{id} = \frac{1}{|C|} \sum_{c \in \mathcal{C}} (\mathcal{L}_{\text{pos}} + \mathcal{L}_{\text{neg}}) \quad (4)$$

Equation 2 effectively pushes the score of each positive event type above the NULL event type and Equation 3 lowers the scores for all negative event types.

For some event types such as "Business:Lay off", the natural language label "lay off" cannot be mapped to a single token. In this case, we add a new token ⟨`lay_off`⟩ and initialize its embeddings as the average of the tokens that compose the original event name.

2.2 Trigger Localization

Trigger localization is the task of finding the trigger offset given a context c and an event type t. Since we already know the event type, we can construct a more informative input by leveraging external knowledge (for instance, from FrameNet) about the event type. For example, in Figure 3, we use the event description from the annotation guidelines to help define the "Start-Position" event type. We can also use a few keywords (example triggers) to serve as the event knowledge. In our experiments we compare the two forms of event knowledge.

Our localization model is a linear chain CRF tagger with only three tags: BIO[3]. In this way, the model parameters are not tied with any event type and can be easily used for transfer.

The probability of a tagged sequence is:

$$p(y|\vec{h}; \theta) = \frac{\exp\left(\sum_i \varphi(y_i|h_i) + \sum_i \psi(y_i|y_{i-1})\right)}{Z} \quad (5)$$

[3]B stands for the beginning of a span, I stands for the inside of a span, and O for outside of any span.

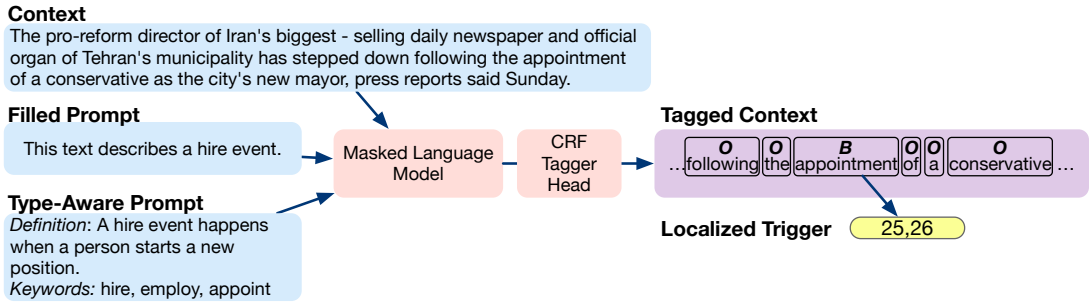

Figure 3: The localization model. The context, filled prompt (from the identification stage), and a *type-aware prompt* are provided as input. The *type-aware prompt* can be the event definition or event keywords. Our model outputs type-free BIO tags for the context which can then be converted into trigger locations.

where $\vec{h}$ is the contextualized embedding vector of the tokens from the masked language model and Z is a normalization factor.

We parameterize the emission scorer $\varphi(y_i|h_i)$ as:

$$\varphi(y_i|h_i) = W_l h_i + \sum_j \alpha_{ij} W_v h_j \quad (6)$$

Both $W_l \in \mathbb{R}^{3\times m}$ and $W_v \in \mathbb{R}^{3\times m}$ map the embeddings to the tag space, serving as an early prediction. Then we fuse the predictions for the token and the other tokens through an attention mechanism with the weight α_{ij} defined as:

$$\alpha_{ij} = \text{Softmax}_j \left(\frac{(W_q h_i)^T W_k h_j}{\sqrt{m}} \right) \quad (7)$$

m is the dimension of the embeddings h and $W_q \in \mathbb{R}^{m\times m}$, $W_k \in \mathbb{R}^{m\times m}$ are learnable parameters.

2.3 Joint Training

In a sense, our identification model captures the probability of the event type given the context $p(t|x)$ and our localization model captures the probability of the token tags given the context and event type: $p(y|t, x)$.

The identification model and the localization model share the same masked language model backbone. Since these two tasks have slightly different inputs, we alternate between sampling batches for each task.

3 Experiments

In the following experiments, we refer to our proposed model as PILED, standing for *Prompt-guided Identify-then-Localize Event Detection*.

Dataset	# Docs	# Sents	# Event types	# Instances
ACE+	599	20,818	33	5,311
FewEvent	-	70,852	100	70,852
MAVEN	4,480	49,873	168	118, 732

Table 1: Dataset statistics.

Datasets We evaluate our model on three datasets, FewEvent (Deng et al., 2020), MAVEN (Wang et al., 2020) and ACE2005[4].

FewEvent is designed to be a few-shot event detection benchmark aggregating data from ACE, TAC-KBP (Ji and Grishman, 2011) and expanding to additional event types related to sports, music, education, etc. from Wikipedia and Freebase. We follow the data split released by (Cong et al., 2021).

MAVEN is the largest human annotated event detection dataset to date, covering 4,480 documents and 168 event types. We use MAVEN for the few-shot setting following (Chen et al., 2021).

ACE2005 is the most widely used dataset for event extraction. For data preprocessing, we follow (Lin et al., 2020) and keep multi-word triggers and pronouns. We denote this version of ACE2005 as ACE+. Since FewEvent has significant data overlap with ACE2005, we do not further experiment with the few-shot setting on ACE 2005.

We present the overall dataset statistics in Table 1. Details of the data splits are available in the Appendix.

Evaluation Metrics For all experiments, we use the event instance precision, recall and micro-F1 score as our major evaluation metrics. An event mention is considered correct if both its type and trigger span are correct.

[4] https://www.ldc.upenn.edu/collaborations/past-projects/ace

Implementation Details We use Roberta (Liu et al., 2019) as our masked language model. For experiments, we match the model size of our baselines for fair comparison: on ACE, we use Roberta-large and on FewEvent and MAVEN, we use Roberta-base. For Roberta-base, we use a batch size of 8 and a learning rate of $2e-5$. For Roberta-large, we use a batch size of 16 and a learning rate of $1e-5$. We set the maximum sequence length to 200 tokens since our predictions are on the sentence-level. For more details, we refer the readers to the Appendix.

3.1 Few Shot Event Detection

For few-shot experiments, instead of following the episode-based setup in PA-CRF (Cong et al., 2021), we use the more standard setup in StructShot (Yang and Katiyar, 2020). In the episode-based setup, each time K labeled instances per event are sampled to form the support set and 1 unlabeled instance is sampled to form the query set. The model is then evaluated on this query set. In our setup, we also sample the K-shot support set, but evaluate our model on all of the remaining unlabeled instances. In expectation, both settings should reach the same performance. Another difference is that since the training set consists of event types that are disjoint to the test set, we do not use the training set at all.

We list our results on the FewEvent dataset in Table 2 and results on the MAVEN dataset in Table 4.

On FewEvent, there is only one event type labeled per sentence, so the identification task is reduced to classification. We compare our identification model to another knowledge-based few-shot event classification model (Shen et al., 2021)[5]. The AKE-BML model uses examples from FrameNet (Baker et al., 1998) as external knowledge. Our event knowledge is exemplified in the choice of the verbalizer for each event type. They also rely on the trigger location for classification whereas we perform identification before localization.

On the localization task, our model can jointly learn from annotation of all event types, giving us a significant advantage (over 16% F1) over sequence labeling models that store "prototype" representations of each event type individually.

[5]Although the title contains "event detection", in the problem definition the task is framed as few-shot classification with known triggers.

On the MAVEN dataset, the increase in event types and the fact that multiple event types can co-occur in the same sentence makes the task more difficult. On the identification task, our prompt-based method can outperform the causal inference enhanced RelNet (Chen et al., 2021; Sung et al., 2018) by 8.5% F1 without having access to the trigger word location. Instead of linking trigger words to a numerical label, our identification model leverages the similarity between the verbalizer and the triggers. For the event detection task (with localization), since no previous work attempted this task, we compare with a token classification baseline that follows the fine-tuning paradigm and adapt a competitive few-shot name tagging model StructShot (Yang and Katiyar, 2020) to our task. Additionally, we show some example predictions in Table 3. The Token Classification baseline has poor performance and high variance due to the sampling of the support set. Due to abundance of 'O' (outside) tags, this baseline also tends to refrain from predicting any event type. The StructShot model is a token-level k-nearest neighbor model with Viterbi decoding. As KNN models are learning-free, the StructShot model performs relatively well under few-shot settings. However, this KNN backbone also limits the model's performance when encountering new triggers as in the case for "study" and "authorized".

3.2 Supervised Event Detection

We report supervised event detection results on the ACE+ dataset in Table 5. We compare with a wide range of existing methods, covering the paradigms of single-task sequence labeling, multi-task learning, question answering and generation. The multitask learning model OneIE enjoys the benefits of joint decoding across related tasks such as entity extraction and relation extraction. Notably, DEGREE (Hsu et al., 2021) also uses event descriptions and keywords as a "type-aware prompt" to guide the generation of the trigger word. However, generation using the entire vocabulary is more challenging than our localization task.

4 Analysis and Discussion

In the previous experiments we showed that our model performs favorably under fully-supervised settings and surpasses previous methods by a large margin under few-shot settings. Here we take a closer look at the design choices in our model.

Task	Model	5 way 5 shot	5 way 10 shot	10 way 5 shot	10 way 10 shot
Identification	AKE-BML (Shen et al., 2021)	88.99	90.10	84.55	86.03
	PILED	**92.69** ± 3.60	**92.18** ± 2.85	**89.60** ± 2.17	**92.72** ± 1.14
Id + Localization	DMBPN (Deng et al., 2020)	37.51	38.14	34.21	35.51
	ProtoNet (Snell et al., 2017)	58.82	61.01	55.04	58.78
	Collapsed CRF (Hou et al., 2020)	59.30	62.77	56.41	59.44
	PA-CRF (Cong et al., 2021)	62.25	64.45	58.48	61.64
	PILED	**79.24** ± 2.61	**81.22** ± 2.30	**81.14** ± 1.93	**83.02** ± 1.35

Table 2: Few-shot event detection results (%) on FewEvent. All results are micro-F1 scores. We report the average and standard deviation across 10 runs.

Context	Model		
	Token Classification	StructShot	PILED
The results of a separate study[Research] indicated that it may have been a larger event, placing the shock in the North Cascades,...	NULL	NULL	Research: study
It was led by the U.S. Marines and U.S Army against the Iraqi insurgents in the city of Fallujah and was authorized[Ratify] by the U.S.-appointed Iraqi Interim Government.	NULL	NULL	Ratify: authorized
With the commencement of the Virgin Tour, a wide-ranging audience, especially young women, thronged to attend, attired[Wearing] in Madonna-inspired clothing.	NULL	Wearing: attired; Wearing: inspired	Wearing: attired
In June 2010, seven Indian nationals who were UCIL employees in 1984, including the former UCIL chairman, were convicted in Bhopal of causing death by negligence and sentenced[Punishments] to two years imprisonment and a fine of about $2,000 each, the maximum punishment allowed by Indian law.	Prison: imprisonment	Punishments: fine	Prison: imprisonment; Punishments: punishment

Table 3: Case studies on the few-shot event detection task. The annotations are marked in the context: the trigger is underlined and the corresponding event name is provided in the bracket. In the last example, we believe that the given annotation is not complete.

Task	Model	Micro F1
Identification	RelNet*	56.0 ± 1.4
	RelNet + Causal*	57.0 ± 0.9
	PILED	**65.5** ± 1.1
Id + Localization	Token Classification	16.3 ± 4.7
	StructShot	40.4 ± 1.0
	PILED	**63.1** ± 1.1

Table 4: Few-shot event detection results (%) on MAVEN. We follow the 45 way 5 shot setting in (Chen et al., 2021) and report the average and standard deviation for 10 runs. Results marked with * are also taken from the aforementioned paper.

4.1 Injecting Event Knowledge

In our model, event knowledge is present in the *verbalizer* in the identification stage and the *type-aware prompt* in the localization stage.

In the previous experiments, we use one manually selected verbalizer per event type. A natural question is whether more verbalizers will help. We use MAVEN for this set of experiments since MAVEN provides alignments between its event types and FrameNet frames. The FrameNet[6] definitions and lexical units can then serve as event knowledge.

When more than one verbalizer is used, we need to aggregate the scores over the verbalizer set. We experiment with 4 different types of aggregation operators: `avg`, `max`, `logsumexp`, `weighted-avg`. The `logsumexp` operator can be seen as a smoothed version of the `max` operator. In the `weighted-avg` operator, the weights of the verbalizers are additional learnable parameters (Hu et al., 2021). As shown in Table 6, in the few-shot setting, using multiple verbalizers can provide 1.5-2% F1 improvement on identification which translates to 1.6-2.2% F1 improvement on the event detection task. In terms of aggregation methods, the `avg` operator is a simple and reliable choice with the best performance and lowest variance. Although the `wavg` operator is more expressive, it is hard to learn good weights with only

[6] https://framenet.icsi.berkeley.edu/fndrupal/frameIndex

Category	Model	Prec	Recall	F1
Sequence labeling	Token Classification	67.1	72.3	69.6
Sequence labeling	Token Classification+CRF	67.8	76.6	71.9
Multitask	OneIE* (Lin et al., 2020)	-	-	<u>72.8</u>
QA	EEQA* (Du and Cardie, 2020)	71.1	73.7	72.4
Generation	Text2Event* (Lu et al., 2021)	71.2	72.5	71.8
Generation	DEGREE* (Hsu et al., 2021)	-	-	72.7
Prompt-based	PILED	70.9	76.1	**73.4**

Table 5: Supervised event detection results (%) on ACE+. The best results are in boldface and the next best results are underlined.* indicates results cited from the original paper.

Agg method	Id F1	Id+Loc F1
`avg`	**67.5** $\pm$ 1.6	**65.3** $\pm$ 1.4
`max`	67.0 $\pm$ 2.2	64.7 $\pm$ 2.2
`logsumexp`	67.0 $\pm$ 1.9	64.7 $\pm$ 1.9
`wavg`	67.4 $\pm$ 1.6	64.9 $\pm$ 1.7

Table 6: Using multiple verbalizers for the 45-way-5-shot event detection on the MAVEN dataset. To balance between frames that have different number of lexical units, we use at most 3 verbalizers. `wavg` stands for `weighted-avg`.

5 examples per event type.

For the type-aware prompt, we consider using the event definition or event keywords and compare it against the baseline of using the filled prompt from the identification stage. As seen in Table 7, the event name alone is relatively informative and adding event keywords can provide an additional 0.8% F1 gain. The definitions from FrameNet are highly abstract, which may undermine their value in assisting event detection.

Event knowledge	Id F1	Loc F1
Event name	64.8 $\pm$ 1.3	62.0 $\pm$ 1.5
Name + Definition	64.8 $\pm$ 1.3	62.3 $\pm$ 1.5
Name + Keywords	**65.5** $\pm$ 1.1	**63.1** $\pm$ 1.1

Table 7: Comparison of using different types of event knowledge to construct the type-aware prompt for localization. The event name is present in the filled prompt. We use at most 3 keywords per event type.

Id Model	Loc Model	Prec	Recall	F1
✓	Full model	70.9	76.1	73.4
✓	Single class CRF	68.3	74.9	71.5
✓	QA	72.5	69.0	70.7
✓	Span Classifier	63.5	78.3	70.1
Enumerate	Full model	54.5	81.3	65.3

Table 8: Model ablations on ACE+.

4.2 Model Design Choices

We design our localization model as an attention-enhanced single-class CRF tagger. However, there are many alternative modeling choices for detecting the trigger offset. Here, we experiment with other common models including the question answering (QA) formulation (Du and Cardie, 2020; Liu et al., 2020) , the span classification formulation (Span Classifier) and the vanilla CRF model as shown in Table 8. For the single-class CRF model, we remove the attention based early-interaction term in Equation 7. In the question answering formulation, we compute the scores of the token being first token in the answer (the answer head) and being the last token in the answer (the answer tail) separately. This simple QA model cannot handle multiple "answers" per sentence, so we extend it to a span classification model where each span is scored independently and assigned a binary label.

Although the span classifier can handle multiple triggers in the same sentence, it suffers from low precision. Compared to the QA model and the span classifier model which score candidate triggers independently, the vanilla CRF model explicitly models the correlation between neighboring tokens, leading to better performance. Additionally, our attention-enhanced CRF layer can further improve upon the vanilla CRF model by 1.9 % F1 points.

One alternative to the *identify-then-localize* framework is to simply enumerate all possible event types and attempt to localize the trigger for them. To verify if the identification step is truly necessary, we compare our two-stage model with a localization-only model that enumerates all possible event types. As shown in the last row of Table 8, this model has high recall at the cost of low precision. Additionally, with N event types in the ontology, this model requires N times training time and inference time.

5 Related Work

5.1 Prompt-Tuning

The pioneer of prompt-tuning is the concept of in-context learning introduced by GPT-3 (Brown et al., 2020), demonstrating the few-shot capability of large pretrained language models. What sets prompt-tuning apart from the widely used fine-tuning approach is that the task specifications (task description or examples) are provided as part of the input. Depending on the format of the prompt, prompt-tuning methods can be divided into cloze-style prompts for classification (Schick and Schütze, 2021a,b) and open-ended prompts for generation (Li and Liang, 2021). Based on the human readability of the prompts, they can be either discrete (Shin et al., 2020), or continuous (Qin and Eisner, 2021). For a more comprehensive view on the work in prompt-tuning, we refer readers to (Liu et al., 2021).

Application-wise, prompt-tuning has been shown to be very successful for classification and generation tasks. There have been some recent attempts to apply prompt-tuning to informative extraction tasks such as named entity recognition (Ding et al., 2021) and relation extraction (Han et al., 2021) but they largely focus on the classification component of these tasks after locating the target spans. To date, we are the first to tailor prompt-learning for the event detection task.

5.2 Low Resource Event Detection

Due to the high cost of annotating event instances, low resource event detection has received much attention in recent years. There are a variety of settings explored, including zero-shot transfer learning (Lyu et al., 2021; Huang et al., 2018), cross-lingual transfer (Subburathinam et al., 2019), inducing event types (Huang et al., 2016; Wang et al., 2021; Huang and Ji, 2020), keyword-based supervision (Zhang et al., 2021) and few-shot learning (Peng et al., 2016; Lai et al., 2020; Shen et al., 2021; Cong et al., 2021; Chen et al., 2021).

Methodology-wise, prototype-based methods (Deng et al., 2020; Zhang et al., 2021; Cong et al., 2021; Shen et al., 2021) have been a popular choice since they were originally developed for few-shot learning. Either starting from keywords (Zhang et al., 2021), definitions (Shen et al., 2021) or examples (Deng et al., 2020; Cong et al., 2021), the key is to learn a good representation for each event type (often referred to as the class prototype) and then predict the event type of the new example using a certain proximity metric to the "prototype".

Another idea is to transfer knowledge from semantic parsers, such as AMR (Wang et al., 2021; Huang et al., 2018) or SRL (Zhang et al., 2021; Lyu et al., 2021) parsers. The event detection task is then converted into the task of finding a mapping between the predicates detected by the semantic parser to event types in the target ontology. Such methods are dependent on the performance of the semantic parsers.

QA-based (Du and Cardie, 2020; Liu et al., 2020) and generation-based methods (Li et al., 2021; Hsu et al., 2021) can also be adapted to the problem since event type information can be incorporated into the input. However, with this flexibility comes a drawback: if a general question such as "What is the trigger?" is asked, the model cannot quickly adapt to new types; if a type-specific question such as "What is the trigger for attack?" is used, the model has to be queried once per possible event type to reach the final answer. For the sake of efficiency, we formulate the identification step as a multi-class classification problem. We also compare our two-stage model's performance with this enumerative approach in Section 4.2.

6 Conclusions and Future Work

In this paper we study event detection under few-shot learning settings. Inspired by cloze prompts that can bridge the gap between pretrained masked language models and a target task through a task description, we extend this idea to event detection by formulating the problem as an *identify-then-localize* procedure. Specifically, we first *identify* the event types present in the context and then find the trigger *location* based on type-specific event knowledge. We show that this approach significantly outperforms existing methods for few-shot event detection, achieving a 16% absolute F1 score gain on FewEvent and 23% gain on MAVEN.

In the process of performing this study, we also realized some caveats in the current few-shot learning evaluation and we hope that more realistic evaluations of few-shot event detection models can be designed in future work. Another interesting extension would be to develop interactive systems where the user can constantly provide feedback to assist the extraction of new event types, especially when the initial examples carry ambiguity.

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *COLING-ACL '98: Proceedings of the Conference*, pages 86–90, Montreal, Canada.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, T. J. Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeff Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *NeurIPS*.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. *EMNLP*.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-Shot Event Detection with Prototypical Amortized Conditional Random Field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40, Online. Association for Computational Linguistics.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. *Proceedings of the 13th International Conference on Web Search and Data Mining*.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juan-Zi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *ArXiv*, abs/2108.10604.

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

Tianyu Gao, Xu Han, Hao Zhu, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2019. FewRel 2.0: Towards more challenging few-shot relation classification. 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 6250–6255, Hong Kong, China. Association for Computational Linguistics.

Xu Han, Weilin Zhao, NNN, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *ArXiv*, abs/2105.11259.

Yutai Hou, Wanxiang Che, Yongkui Lai, Zhihan Zhou, Yijia Liu, Han Liu, and Ting Liu. 2020. Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1381–1393, Online. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2021. Degree: A data-efficient generative event extraction model.

Shengding Hu, NNN, Huadong Wang, Zhiyuan Liu, Juan-Zi Li, and Maosong Sun. 2021. Knowledgeable prompt-tuning: Incorporating knowledge into prompt verbalizer for text classification. *ArXiv*, abs/2108.02035.

Lifu Huang, Taylor Cassidy, Xiaocheng Feng, Heng Ji, Clare R. 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, Berlin, Germany. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proc. ACL2011*.

Viet Dac Lai, Thien Huu Nguyen, and Franck Dernoncourt. 2020. Extensively matching for few-shot learning event detection. In *Proceedings of the First Joint Workshop on Narrative Understanding, Storylines, and Events*, pages 38–45, Online. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021. 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, Online. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597, Online. Association for Computational Linguistics.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7999–8009, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ArXiv*, abs/2107.13586.

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

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. 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 2795–2806, Online. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 392–402, Austin, Texas. Association for Computational Linguistics.

Guanghui Qin and Jason Eisner. 2021. Learning how to ask: Querying LMs with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5203–5212, Online. Association for Computational Linguistics.

O. Mahamane Sani Sabo, Yanai Elazar, Yoav Goldberg, and Ido Dagan. 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. *Transactions of the Association for Computational Linguistics*, 9:691–706.

Timo Schick and Hinrich Schütze. 2021a. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Timo Schick and Hinrich Schütze. 2021b. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Shirong Shen, Tongtong Wu, Guilin Qi, Yuan-Fang Li, Gholamreza Haffari, and Sheng Bi. 2021. Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2417–2429, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *NIPS*.

Ananya Subburathinam, Di Lu, Heng Ji, Jonathan May, Shih-Fu Chang, Avirup Sil, and Clare Voss. 2019. Cross-lingual structure transfer for relation and event extraction. 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 313–325, Hong Kong, China. Association for Computational Linguistics.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip H. S. Torr, and Timothy M. Hospedales. 2018. Learning to compare: Relation network for few-shot learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1199–1208.

Oriol Vinyals, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. 2016. Matching networks for one shot learning. In *NIPS*.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations.

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 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Xiaozhi Wang, Ziqi Wang, Xu Han, Wangyi Jiang, Rong Han, Zhiyuan Liu, Juanzi Li, Peng Li, Yankai Lin, and Jie Zhou. 2020. MAVEN: A Massive General Domain Event Detection Dataset. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1652–1671, Online. Association for Computational Linguistics.

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. 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 6283–6297, Online. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021. Zero-shot Label-aware Event Trigger and Argument Classification. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1331–1340, Online. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jinke Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. In *AAAI*.

A Dataset Details

For FewEvent, we use the data split from (Cong et al., 2021) and use 80 event types as the training set, 10 event types as the dev set and the remaining 10 event types as the test set. In the data provided, sentences are organized by event type and each sentence only has one event mention annotation.

	Train	Dev	Test
# Types	80	10	10
# Sents	67,982	2,173	697

Table 9: Data split for FewEvent.

In the N-way K-shot experiments, we randomly sample N event types from the test set and then sample K labeled instances of that event type for training.

For MAVEN, we follow the data split by (Chen et al., 2021) and use the sentences containing the most frequent 120 event types as the training set. The sentences containing the remaining 45 event type are then split into half as the dev and test set. We use the same random seed as (Chen et al., 2021) to ensure the same split.

	Train	Dev	Test
# Types	125	45	45
# Sents	86, 551	1,532	1,555
# Events	287, 516	1,741	1,806

Table 10: Data split for MAVEN.

For ACE, we use the data split in (Lin et al., 2020). The same 33 event types are shared in the training, dev and test set.

	Train	Dev	Test
# Sents	19, 240	902	676
# Events	4,419	468	424

Table 11: Data split for ACE+.

B Additional Case Studies

We show some additional case studies in Table 12. In the first case, our model can differentiate between similar event types such as “Commerce_pay" and “Cost" while StructShot identifies both as “Cost". In the second case, we show an event type that is very specific: “Bearing_arms" is almost always triggered by the word “armed" and our model handles it well. The “Filling" event type, on the contrary, is very general, and according to FrameNet, can describe “filling containers and covering areas with some thing, things or substance". This shows that our model can generalize to rare triggers like “inundated". In the last example, we show a failure case of our model. As the event “Cost" is often triggered by verbs, the model fails to recognize that “prices" is also contained in the event type.

C Model Hyperparameters

For the experiments on ACE+, we used the settings and hyperparameters as shown in Table 13.

Context	Model		
	Token Classification	StructShot	PILED
Kuwait and Saudi Arabia paid[Commerce_pay] around US$32 billion of the US$60 billion cost[Cost].	NULL	Cost: paid; Cost: cost	Commerce_pay: paid; Cost: cost
The conflict has lasted for over 39 years, making it the second longest internal conflict in the history of Latin America, after the Colombian armed[Bearing_arms] conflict.	NULL	NULL	Bearing_arms: armed
Strong winds and heavy rainfall inundated[Filling] streets, residences, and fields, and also toppled chimneys, fences, and cracked windows across the region.	NULL	NULL	Filling: inundated
Official ticket prices[Cost] through Ticketmaster ranged from $71.70 to $439.90.	NULL	NULL	NULL

Table 12: Extra case studies on the few-shot event detection task. The annotations are marked in the context: the trigger is underlined and the corresponding event name is provided in the bracket.

Parameter	Value
Encoder	Roberta-large
Max seq len	200
Batch size	16
Learning rate	$1e-5$
Learning rate schedule	Linear
Weight decay	$1e-5$
Warmup steps	1000
Epochs	10
Adam ϵ	$1e-8$
Gradient clipping	1.0

Table 13: ACE+ hyperparameters

For all few-shot experiments, we use the parameters listed in Table 14.

Parameter	Value
Encoder	Roberta-base
Max seq len	200
Batch size	8
Learning rate	$2e-5$
Learning rate schedule	Linear
Weight decay	$1e-5$
Warmup steps	0
Epochs	20
Adam ϵ	$1e-8$
Gradient clipping	1.0

Table 14: Few-shot experiment hyperparameters.

D Discussion on Few-shot Learning Datasets

Few-shot learning for event detection was largely inspired by the few-shot classification work in computer vision literature (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018) which assumes that images are sampled independently under the N-way K-shot setting. However, this assumption does not directly transfer to context-dependent tasks such as event detection: the distribution of event types heavily depends on the document and is far from i.i.d. in practice. This sampling procedure also leads to the absence of the NULL class (sentences without any event mentions), which is often abundant in real documents.

This data discrepancy has received some attention in other tasks such as relation extraction (Gao et al., 2019; Sabo et al., 2021) but is under-explored for event detection. For example, FewEvent instances only contain one event type per sentence and do not include NULL class examples. Sentences from MAVEN may contain multiple event types but also exclude the case of NULL. Thus, many previous works in few-shot event detection simply design their model to be a K-way classifier. ACE, the dataset which we use for supervised event detection, contains all these cases and the events follow a natural distribution but the small number of event types makes it less attractive to use as a few-shot benchmark. Our model PILED is capable of handling these cases, as exemplified by our performance on ACE, but such abilities were not put to test on the current few-shot datasets. As a result, we would like to remind readers of the possible inflation of few-shot performance on current benchmarks and call for future research on setting up better evaluation.