Learning with Partial Annotations for Event Detection

Jian Liu[1], Dianbo Sui[2], Kang Liu[3], Haoyan Liu[4] and Zhe Zhao[4]
[1] Beijing Jiaotong University [2] Harbin Institute of Technology
[3] The Laboratory of Cognition and Decision Intelligence for Complex Systems
Institute of Automation, Chinese Academy of Sciences
[4] Tencent AI Lab
jianliu@bjtu.edu.cn; dianbosui@hit.edu.cn; kliu@nlpr.ia.ac.cn
{haoyanliu, nlpzhezhao}@tencent.com

Abstract

Event detection (ED) seeks to discover and classify event instances in plain texts. Previous methods for ED typically adopt supervised learning, requiring fully labeled and high-quality training data. However, in a real-world application, we may not obtain clean training data but only partially labeled one, which could substantially impede the learning process. In this work, we conduct a seminal study for learning with partial annotations for ED. We propose a new trigger localization formulation using contrastive learning to distinguish ground-truth triggers from contexts, showing a decent robustness for addressing partial annotation noise. Impressively, in an extreme scenario where more than 90% of events are unlabeled, our approach achieves an F1 score of over 60%. In addition, we re-annotate and make available two fully annotated subsets of ACE 2005 to serve as an unbiased benchmark for event detection. We hope our approach and data will inspire future studies on this vital yet understudied problem.

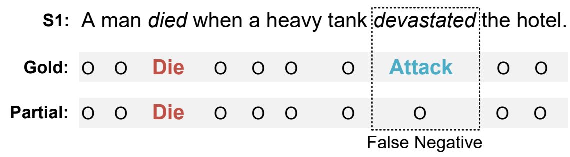

Figure 1: The partial annotation issue in ED. The Gold row indicates ground-truth labels; the Partial row indicates the partial annotation case we address in this study, where the *devastated* event is not labeled.

1 Introduction

Deep learning models have shown impressive performance in event detection (ED) since large amounts of event data have become available (Chen et al., 2015; Nguyen and Grishman, 2015). However, such models require fully labelled and high-quality data — in practice, we cannot ensure that every event is identified, and as a result, we often face the *partial annotation* issue, as depicted in Figure 1. We show a high rate of partial annotation in real-world event datasets. For example, in ACE 2005, which is widely used as a benchmark for ED evaluation (Christopher Walker and Maeda, 2006), nearly 20% of events are not labelled (see Table 2). Using a partially labelled dataset as a fully labelled one for training runs the risk of mis-training on false negatives, and using a partially labelled dataset for evaluation biases comparison. However, this issue is still understudied in the existing literature (Liu, 2018; Liu et al., 2020b).

In this work, we present a seminal study of learning with partial annotations for ED, with contributions in methodology, data, and practical applications. In our method, to reduce the risk of mis-training on false negatives, we propose a contrastive learning framework (Chopra et al., 2005; Chen et al., 2020) to distinguish ground-truth triggers from contexts, which is shown to be more tolerant of partial annotation noise than the traditional hard classification paradigm (Ji and Grishman, 2008; Chen et al., 2015). In addition, to succeed in the partial annotation scenario, we augment the model with a self-correction regime to recognize false negatives during the training stage.

Figure 2 visualizes the core of our method, which is a de facto *trigger localization* formulation that uses sentence-wise normalization (prompted by event types) to find event triggers. Compared to hard classification methods that add up individual losses (as shown at the top of Figure 2), our approach instead forms a contrastive learning paradigm by raising the scores of ground-truth triggers while lowering the scores of context words. As a result, even with a significant number of false negatives in training, it can still maintain a good separation between triggers and contexts (§ 6.1). In addition, we suggest that adding a margin softmax (Wang et al., 2018) with a Gaussian-based distance regularization can further improve learning.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 508–523
July 9-14, 2023 ©2023 Association for Computational Linguistics

In addition to the noise-tolerance mechanism described above, we propose a self-correction regime with the motive that when a model recognizes a false negative with high confidence, it should correct its labels for the subsequent training stage. Nevertheless, modeling the confidence of deep learning models is challenging since their predictions are poorly calibrated (i.e., a model often outputs a high prediction probability even if the prediction is incorrect (Guo et al., 2017)). To address this issue, we propose an uncertainty-guided retraining mechanism based on MC-Dropout (Gal and Ghahramani, 2016), which can output prediction confidence to guide the self-correction process. We explain the relationship between this paradigm and an expectation-maximization (EM) framework.

In addition to the methodology contribution, we re-annotate and release the ACE 2005 development and test sets as a data contribution. On the revised benchmark (and an extra MAVEN (Wang et al., 2020) benchmark), we demonstrate the impressive performance of our models — in particular, even in an extreme case with 90% of events unlabeled, our approach achieves more than 60% in F1, yielding a 40% definite improvement over previous methods. In addition to simulation tests, we also conduct a real-world annotation test on WikiEvents (Li et al., 2021a), where the results suggest the practical applicability of our approach.

Contributions. Our contributions are three-fold: (i) To the best of our knowledge, this is the first work addressing the potential partial annotation issue in ED, which may spark further research interest. (ii) We highlight a new learning paradigm for ED based on a trigger localization formulation and show that it works effectively with a wide range of partial annotation settings. (iii) We re-annotated the ACE 2005 development and test datasets and released them to the community to serve as an unbiased benchmark. (IV) In addition to simulation experiments, we conduct real-world annotation experiments to validate the effectiveness of our approach for practical use.

2 Related Work

ED and the Partial Annotation Issue. Event detection (ED) is a crucial subtask of event extraction that aims to identify event instances in texts (Grishman, 1997; Ahn, 2006). The existing ED methods can be divided as feature-based (Ahn, 2006; Li et al., 2013; Liao and Grishman, 2010; Hong et al., 2011) and deep learning-based (Chen et al., 2015; Nguyen and Grishman, 2015; Nguyen et al., 2016; Liu et al., 2018a; Feng et al., 2016; Chen et al., 2018; Yang et al., 2019; Liu et al., 2020a; Du and Cardie, 2020; Lu et al., 2021; Liu et al., 2019a), and there has been a growing interest in applying these methods to specific scenarios (Liu et al., 2019b, 2022b,a). Nevertheless, most of such methods adopt supervised learning and assume the availability of clean datasets. To date, only a few studies have considered the partial annotation issue in ED: Liu et al. (2020b) identify several unlabeled cases in the ACE test set for error analysis; Liu (2018), in the PhD proposal, suggest that the Chinese portion of ACE 2005 is partially labeled. Unfortunately, neither work stands in a methodology perspective for addressing the issue. Our research, on the other hand, introduces a solution for learning with partial annotations. Our trigger localization formulation also relates to using prompts for event information extraction (Wang et al., 2022a; Hsu et al., 2022; Liu et al., 2022c; Wang et al., 2022b), but different from them focusing on improving the overall performance, our work stands in a point addressing the partial annotation issue.

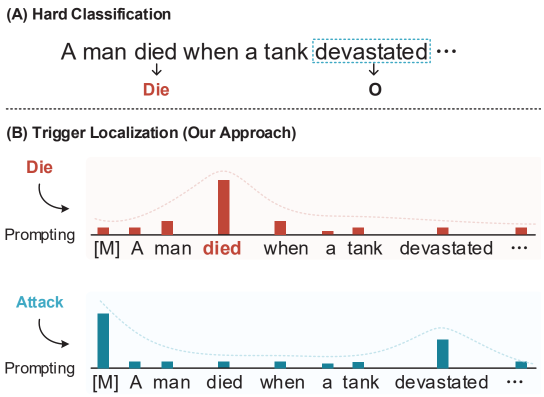

Figure 2: A comparison of the hard classification paradigm (A) and our trigger localization paradigm (B) for ED. [M] is a "no-event-existing" indicator.

Learning with Partial Annotations. Learning with partial annotations, also known as positive and unlabeled learning (Li et al., 2009), is an important problem in machine learning community (Elkan and Noto, 2008; Liu et al., 2002, 2003, 2005). In the domain of natural language processing (NLP), researchers have examined a number of tasks including named entity recognition (NER) (Jie et al., 2019; Mayhew et al., 2019; Peng et al., 2019), Chinese word segmentation (Yang and Vozila, 2014),

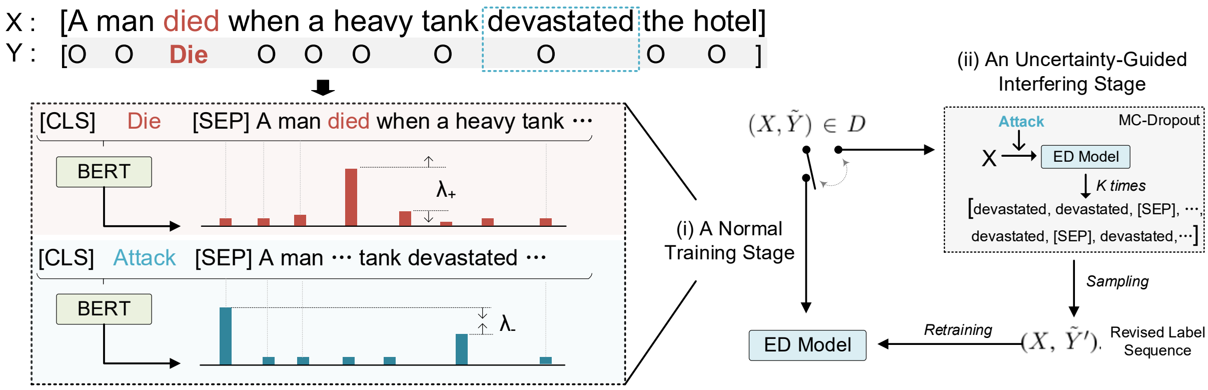

Figure 3: The overview of our approach. Left: the model training process based on margin softmax. Right: the uncertainty-guided retraining mechanism. D designates the original training dataset.

and others (Tsuboi et al., 2008). The efforts for NER relate the most to our work, where a seminal work (Jie et al., 2019) treats the labels of negative instances as latent variables and infers them using partial Conditional Random Fields (Bellare and McCallum, 2007). Later works have devised down-weighing mechanisms (Mayhew et al., 2019), confidence estimation methods (Liu et al., 2021), and negative sampling (Li et al., 2021b) for learning. In this study, we offer a new trigger localization formulation for the task of ED and demonstrate promising results in a wide range of partial annotation settings.

3 Proposed Method

Let $X = [w_1, \cdots, w_N]$ be a sentence with N words and $Y = [y_1, \cdots, y_N]$ be the ground-truth event label sequence, where $y_i \in \mathcal{T} \cup \{\text{O}\}$ is the event label of w_i (Here $\mathcal{T}$ is a set of all event types and O is a special type for non-trigger words). Then the partial annotation issue can be formulated as: due to the neglect of human annotators, some $y_i \neq$ O are not identified, and this results in a partial annotation sequence $\tilde{Y} = [\tilde{y}_1, \cdots, \tilde{y}_N]$. Clearly, directly training a model on $(X, \tilde{Y})$ risks outputting a noisy detector. Here we propose a new learning framework to address this issue (as shown in Figure 3), which consists of a noise-tolerant learning mechanism with margin softmax (§ 3.2) and uncertainty-guided retraining mechanism (§ 3.3).

3.1 Input Representation

Given a sentence X, for each event type $t \in \mathcal{T}$, we learn their joint representations for further processing. Particularly, we concatenate t and X as the input[1] of a BERT encoder (Devlin et al., 2019):

$$\texttt{[CLS]}\ t\ \texttt{[SEP]}\ \overbrace{w_1\ w_2\ \cdots\ w_N}^{\text{The sentence } X}$$

and consider the output of BERT to be the joint representations, denoted as $\boldsymbol{H}_{(t,X)} \in \mathcal{R}^{M \times d}$, with M being the length of the input sequence[2] and d being the dimension of BERT.

3.2 Noise-Tolerant Learning via Margin Softmax

Based on $\boldsymbol{H}_{(t,X)}$, we next locate event triggers of type t in the sentence. This can be achieved using sentence-level softmax, and here we introduce a *margin softmax* (Wang et al., 2018) to better address partial annotations. Specifically, we first map $\boldsymbol{H}_{(t,X)}$ to a score vector $\boldsymbol{s}^t \in \mathcal{R}^M$ using $\boldsymbol{s}^t = \boldsymbol{H}_{(t,X)}\ \boldsymbol{w}$, where $\boldsymbol{w} \in \mathcal{R}^d$ is a shared vector parameter, and then we distinguish between the following two cases for learning:

Case 1. A positive instance (i.e., a labeled trigger) of type t is found in the label sequence $\tilde{Y}$ (If many triggers are found, we address each one individually). Assume j is the labeled trigger's index. Here we employ a *positive margin* λ_+ and maximize the following objective:

$$p_+(X, \tilde{Y}, j) = \frac{\exp(s^t_{(j)} - \lambda_+)}{\exp(s^t_{(j)} - \lambda_+) + \sum_{m \neq j}^{M} \exp(s^t_{(m)})} \quad (1)$$

where $s^t_{(j)}$ denotes the j^{th} word's score in the score vector $\boldsymbol{s}^t$. This objective encourages a margin of at least $\lambda+$ (Wang et al., 2018) in scores of triggers

[1] [CLS] and [SEP] are special tokens used in BERT.

[2] A word in BERT may broke down into many subwords (Sennrich et al., 2016), and here we only consider the first subword to make $\boldsymbol{H}_{(t,X)}$ have the same length as the input.

and context words, which therefore makes ground-truth triggers more separable. Note in this case, a sentence may contain "hidden" false negatives. Motivated by the fact that triggers are generally sparsely distributed (Lin et al., 2018), we employ a Gaussian regularization to reduce the penalty. Particularly, we obtain a new score vector $\hat{\boldsymbol{s}}^t$ with:

$$\hat{s}^t_{(m)} = s^t_{(m)} \times \mathcal{N}(|m - j|) \tag{2}$$

where $\mathcal{N}(\cdot)$ indicates the standard univariate Gaussian density function, and $|m - j|$ is the distance between the m^{th} word w_m and the labeled trigger. This new score vector $\hat{\boldsymbol{s}}^t$ put small weights on words far away from the labeled trigger and is shown to marginally improve learning (§ 6.2).

Case 2. There is no trigger of type t found in $\tilde{Y}$. In this case, we use the `[SEP]` token as a "no-event-existing" indicator and optimize to give it the highest score. It should be noted, however, that such a case may contain false negatives. To address them, we introduce a *negative margin* λ_-[3] to reduce the penalty:

$$p_-(X, \tilde{Y}) = \frac{\exp(s^t_\triangle + \lambda_-)}{\exp(s^t_\triangle + \lambda_-) + \sum^M_{m \neq \triangle} \exp(s^t_{(m)})} \tag{3}$$

where $\triangle$ indicates the index of `[SEP]`. In this way, the negative margin λ_- instead loosens the gap between the indicator `[SEP]` and other words. As a result, the model is more forgiving of situations when certain words score higher than the "no event exists" indicator `[SEP]`.

Training and Testing Protocols. The overall loss function for learning is:

$$\begin{aligned} \mathcal{L} = - \sum_{(X,\tilde{Y}) \in D} \sum_{t \in \mathcal{T}} [\, \delta_t \times \sum_{j: y_j = t} \log p_+(X, \tilde{Y}, j) \, + \\ (1 - \delta_t) \times \log p_-(X, \tilde{Y}) \,] \end{aligned} \tag{4}$$

where $(X, \tilde{Y}) \in D$ ranges over each instance in the training set D; t enumerates each event type; δ_t is a Dirac delta function:

$$\delta_t = \begin{cases} 1 & \text{If a trigger of type } t \text{ is found (Case 1)} \\ 0 & \text{Otherwise (Case 2)} \end{cases} \tag{5}$$

In the inference stage, given a test sentence X, we compute a normalized probability vector $\boldsymbol{p}^t$:

$$\boldsymbol{p}^t = \text{softmax}(\boldsymbol{s}^t) \tag{6}$$

[3]Note that λ_- has a positive value. We name it "negative margin" just to distinguish from the positive margin $\lambda+$.

Algorithm 1: Uncertainty-guided retraining regime

Input : The training dataset $D = \{X_i, \tilde{Y}_i\}^n_{i=1}$;
Output : The optimal model parameter Θ;
while *not convergence* **do**
  Sample a training example $(X, \tilde{Y})$ from $\mathcal{D}$;
  if *It is a burn-in or a normal training stage* **then**
    Update Θ on $(X, \tilde{Y})$ using Equation (4)
  else
    Build an uncertainty-regularized label sequence $\tilde{Y}'$ with MC-Dropout (§ 3.3);
    Update Θ on $(X, \tilde{Y}')$ using Equation (4)
  end if
end while

and then compose a set for event triggers of type t as: $\{w_i \mid p^t_{(i)} > \tau; i \neq_\triangle\}$, where τ is a threshold defined as 1 / N, with N being the length of the sentence (namely, when the predictive probability of a token is above a uniform distribution, we consider it as a trigger).

3.3 Uncertainty-Guided Retraining Regime

In addition to the noise-tolerant learning paradigm, we also design an *uncertainty-guided retraining* mechanism, in which we correct the potential labels for optimization (Algorithm 1).

Assume $(X, \tilde{Y})$ is a training example. In the uncertainty interfering stage, we assume that X is an unlabeled sentence and re-predict the event label sequence using the current model. We use Monte Carlo Dropout (MC-Dropout) (Gal and Ghahramani, 2016) to assess the model's uncertainty on the prediction. Particularly, for each event type t, we predict the event triggers K times with dropout layers activated. Assume the resulting prediction set is $\{q_i\}^K_{i=1}$, where q_i is the i^{th} prediction and $N(q_i)$ is its frequency[4]. We then create a categorical distribution using $N(q_i)/K$ as parameters and sample out a prediction from the categorical distribution as the predict result (This benefits predictions that the model is more confident in). We finally convert the prediction as a label sequence Y' and train a model on $(X, \tilde{Y}')$. We alternate between this uncertainty interfering stage and a standard training stage after several burn-in steps.

Connection to EM Algorithm. Intuitively, our approach can be viewed as an expectation maximization (EM) algorithm (Dempster et al., 1977) using MC-Dropout to approximate the posterior.

[4]For instance, for the sentence shown in Figure 3, if we consider the Attack type and set K = 5, we may get a prediction set: {`[SEP]`, devastated, `[SEP]`, devastated, devastated}. In this case, N([SEP]) = 2 and N(devastated) = 3.

Denote the original log-likelihood function as $\log \mathcal{L}(\Theta; \mathrm{D})$, where Θ and D indicate the model parameter and partially labeled data respectively. Let $\Theta^{(t)}$ be the parameter at the t^{th} iteration. We can view our method as introducing a hidden variable Z to represent the labels of false negatives and then alternating between two steps: (i) An expectation (E) step, which uses the category distribution generated by MC-Dropout as an approximate of the intractable posterior $\mathrm{p}(\mathrm{Z}|\mathrm{D}, \Theta^{(t)})$. (ii) A maximization (M) step, which maximizes the expectation $\mathrm{E}_{\mathrm{p}(\mathrm{Z}|\mathrm{D},\Theta^{(t)})}[\log \mathcal{L}(\Theta; \mathrm{D}, \mathrm{Z})]$ for optimization.

4 Experimental Setups

Datasets. We conduct our experiments on ACE 2005 and MAVEN[5] (Wang et al., 2020), with data statistics shown in Table 1. In light of the partial annotation issue in ACE, we re-annotate its development and test sets, using a method combining automatic potential case identification and human validation (The details are shown in Appendix A). To facilitate a fine-grained analysis, we also split up all potential cases into two categories: (i) a *challenge* set, which consists of unlabeled words where more than half of the ED models predict that they act as triggers, and (ii) a *control* set, which consists of unlabeled words where fewer than half of the ED models predict that they act as triggers. Table 2 gives the final results, indicating the partial annotation issue is crucial — for instance, on the test set the unlabeled ratio is 19.3%.

Implementations. In our approach, we use `BERT-large` architecture for ACE 2005 (Lin et al., 2020; Nguyen et al., 2021), and `BERT-base` for MAVEN (Wang et al., 2020). As for hyper-parameters, the batch size is set to 10 for ACE 2005 and 20 for MAVEN respectively, chosen from [2, 5, 10, 20, 30]. The learning rate is set at 1e-5 for both datasets, chosen from [5e-5, 1e-5, 5e-6, 1e-6]. In the margin softmax regime, the *positive margin* λ_+ is set to 10, and the *negative margin* λ_- is set to 1; these values are chosen from [0.1, 0.5, 1, 5, 10, 50, 100]. In the uncertainty-guided retraining mechanism, the number of prediction times K is empirically set to 20 for a trade-off between speed and efficiency. We release the data and the code at `https://github.com/jianliu-ml/partialED`.

[5] It should be noted that MAVEN provides a candidate trigger set for prediction, so the evaluation problem caused by partial annotation on this dataset is not a concern.

	Data Split	# Doc.	# Sent.	# Word	# Trigger
ACE	Training set	529	17,172	267,959	4,420
	Dev. set	30	923	18,246	505 (558)
	Test set	40	832	19,061	424 (506)
MV	Training set	2,913	32,431	832,186	77,993
	Dev. set	710	8,042	204,556	18,904
	Test set	857	9,400	238,902	21,835

Table 1: Data statistics of ACE and MAVEN (NV). Numbers in parentheses are re-annotation results.

	Split	# Potential	# Validated	UL Rate
Dev. Set	Challenge	78	34 (43.6%)	6.7%
	Control	34	19 (55.9%)	3.8%
	Total	112	53 (47.3%)	10.5%
Test Set	Challenge	86	51 (59.3%)	12.0%
	Control	50	31 (62.0%)	7.3%
	Total	136	82 (60.2%)	19.3%

Table 2: Details of the revised ACE subsets. "UL Rate" indicates the ratio of unlabeled cases to labeled ones.

Evaluation Settings. We investigate three evaluation settings: (i) A full training setting, in which we use the original training set for learning. Yet, because the original training set inevitably contains unlabeled events, this setting is still a partial learning setting. (ii) A data removal setting, in which we exclude a portion of events from the training setting to study whether the performance drop is caused by a degraded number of positive examples. (iii) A data masking setting, in which we remove the labeling information of some events (by replacing their labels to O) to simulate a more serious partial annotation scene. We use precision (P), recall (R), and F1 as evaluation metrics following previous studies (Ji and Grishman, 2008; Li et al., 2013), and to against randomness, we report experimental results based on a 5-run average.

Baselines. We compare our approach to supervised and partial learning methods. For ACE 2005, we consider the following supervised learning methods: Hybrid (Feng et al., 2016), which combines Recurrent Neural Networks and Convolutional Neural Networks; SeqBERT (Yang et al., 2019), which introduces BERT representations; BERTQA (Du and Cardie, 2020; Liu et al., 2020a), which frames ED as a question answering problem; OneIE (Lin et al., 2020), which uses Graph Neural Networks to learn document-level clues; FourIE (Nguyen et al., 2021), which uses an interaction network to combine four information extraction tasks jointly. For MAVEN, we consider DMBERT (Wang et al., 2019) and BERT-CRF (Wang et al.,

Method	Test Set (O)			Test Set (R)		
	P	R	F1	P	R	F1
Hybrid† (2016)	71.4	71.3	71.4	74.4	72.2	73.3
SeqBERT† (2019)	72.5	72.1	72.3	74.1	73.5	73.8
BERTQA (2020)	71.1	73.7	72.4	74.5	74.5	74.5
OneIE (2020)	74.9	**74.5**	74.7	75.9	74.7	75.3
FourIE† (2021)	**75.7**	74.1	**74.9**	76.0	74.6	75.3
HiddenCRF (2019)	68.4	74.5	71.3	75.7	75.5	75.6
NegSPL (2021b)	70.1	74.0	72.0	75.5	75.5	75.5
Self-Pu (2020)	71.1	71.0	71.1	75.7	74.8	75.2
PromptLoc (ours)	73.6	74.2	73.9	**76.4**	**76.8**	**76.6***

Table 3: Results on ACE 2005, where O and R denote the original and revised sets; † signifies our re-implementations and * is a significance test at p = 0.05.

Method	Dev. Set			Test Set		
	P	R	F1	P	R	F1
Hybrid (2016)	62.9	67.2	65.0	63.7	67.0	65.3
OneIE (2021)	64.0	69.0	66.4	64.5	69.3	66.8
BERTQA (2020)	63.8	69.0	66.3	64.9	69.1	66.9
DMBERT (2019)	64.6	70.1	67.2	62.7	72.3	67.1
BERT-CRF (2020)	65.7	68.8	67.2	65.0	70.9	67.8
HiddenCRF (2019)	66.3	68.5	67.4	64.4	72.3	68.1
NegSPL (2021b)	65.6	68.7	67.1	64.9	71.9	68.2
Self-Pu (2020)	66.3	68.0	67.0	64.3	72.3	68.0
PromptLoc (ours)	**67.8**	**69.2**	**68.5***	**65.4**	**72.8**	**68.9***

Table 4: Results on MAVEN. * denotes a significance test with a randomly paired test at p = 0.05.

2020) as baselines. We consider the following partial learning methods: (1) HiddenCRF (Jie et al., 2019), which treats missing labels as latent variables and infers them using a CRF model (We follow the original paper and use SeqBERT for parameter initialization); (2) NegSPL (Li et al., 2021b), which applies negative sampling for learning and shows good results on NER (we use the same strategy to tune the sampling hyper-parameter). (3) Self-Pu (Chen et al., 2020), which is a self-training boosted method for general positive and unlabeled learning. Our approach is denoted by PromptLoc.

5 Experimental Results

Results in the Full Training Setting. Tables 3 and 4 show results in the full training setting. Accordingly, our method achieves the best F1 on the clean ACE test set and the MAVEN development/test set, suggesting its efficacy. Comparing the results on the original ACE test set is interesting: our method has lower precision than other methods, but when applied to the revised set, the precision is greatly boosted — this implies that our

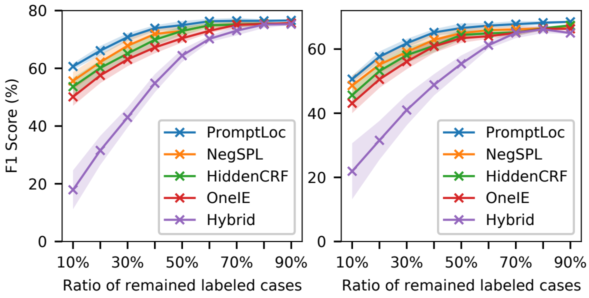

Figure 4: Results on ACE 2005 (left) and MAVEN (right) in the data removal setting.

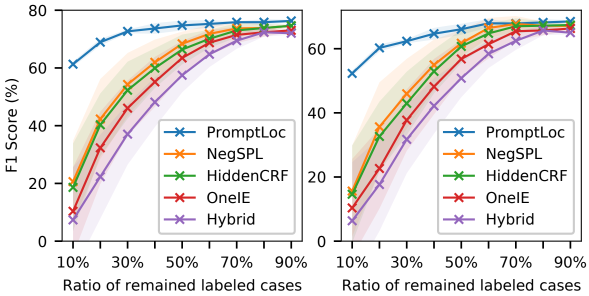

Figure 5: Results on ACE 2005 (left) and MAVEN (right) in the data masking setting.

model does predict triggers not annotated in the original test set. Lastly, partial learning approaches generally outperform supervised methods, showing that the partial annotation issue is a practical concern to be addressed in the ED task.

Results in the Data Removal Setting. Finally, we consider the data removal setting to study the impact of a lack of positive examples, and we show results in Figure 4. According to the results, while our model consistently outperforms others, the gap is small, implying that a reduced number of positive instances is not a major factor impeding learning, especially when there are relatively abundant training examples (e.g., $p > 60\%$) or the pre-trained language models are applied (It does have a significant impact on non-BERT models e.g., Hybrid).

Results in the Data Masking Setting. We then consider the data masking setting and we show results[6] in Figure 5. Here p denotes the ratio of *remaining* examples (i.e., we mask the labels of 1 - p events). According to the results, our approach outperforms prior methods by significant margins. For example, on the ACE 2005, when 70% of triggers are masked ($p = 30\%$), our approach obtains 70% in F1, outperforming previous methods by 30% in F1;

[6] We use the development set for MAVEN because the official site has a submission limit of only 5 per day.

Method	Setting	P	R	F1
Trigger Local.	Clean	73.1	72.9	73.0
	Argmax	70.5	70.5	70.5
	Adaptive τ	68.7	72.8	70.7
Hard Class.	Clean	66.7	76.0	71.0
	Argmax	60.3	42.5	49.8
	Adaptive τ	58.6	43.8	50.1

Table 5: Results of comparing trigger localization and hard classification in extreme partial annotation scene.

when 90% are masked ($p = 10\%$), our approach still achieves 60% in F1, yet previous methods achieves only 20% in F1. Another interesting finding is that our approach yields better results than in the data removal setting (+2.4% and +1.7% in F1 on ACE 2005 and MAVEN). This directly demonstrates our approach's ability to learn from unlabeled events.

6 Qualitative Analysis

6.1 Insights of the Formulation

We conduct a sanity check experiment to understand why our trigger localization paradigm works. First, we randomly select an event type and collect N sentences (200 in our experiments) with events of this type. Then, we create two training examples from each sentence: one with original labels and one with all labels masked — this results in a highly mislabeled dataset. Finally, we train two models — one for trigger localization and the other for hard classification — and evaluate them on a leave-out test set. Table 5 gives the results, where we note that even in this extreme partial annotation scene, our trigger localization paradigm performs well, yielding 70.5% in F1 compared to 73.0% in F1 using clean dataset for training. The hard classification based approach, on the other hand, behaves poorly, yielding a drop of 30% in F1.

In Figure 6, we visualize the learned probabilities of two models on ground-truth triggers and contexts. According to the results, our method can maintain a separation between ground-truth triggers and context words in this extreme partial annotation scene. However, the hard classification-based model is very sensitive to partial annotation noise and can not obtain a clear boundary between the ground-truth triggers and context words. For the above reason, incorporating an adaptive τ has little effect on the performance (Table 5).

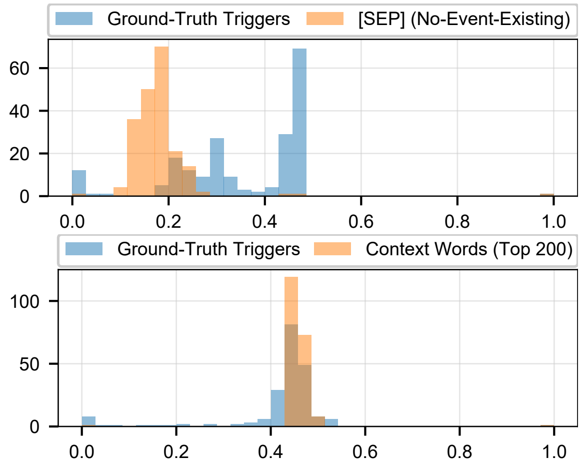

Figure 6: The learned probability distribution.

Method	ACE 2005 10%	ACE 2005 50%	ACE 2005 100%	MAVEN 10%	MAVEN 50%	MAVEN 100%
PromptLoc (SM)	58.0	73.6	76.0	50.1	65.3	67.5
+ λ_+	58.7	74.1	76.0	51.2	65.7	67.7
+ λ_-	**59.4**	**74.3**	**76.5**	51.7	**66.0**	**68.3**
+ Gau. Reg.	59.1	73.9	76.2	**52.0**	65.4	68.0
PromptLoc (Full)	61.3	74.5	76.6	52.3	66.1	68.5
w/o UNT	57.2 $\downarrow$	72.5 $\downarrow$	75.5 $\downarrow$	50.1 $\downarrow$	63.4 $\downarrow$	67.2 $\downarrow$
DirectPred	52.5	70.1	74.6	45.9	61.4	66.9
BoostLearn	44.2	68.1	75.0	41.1	58.7	66.2
OneIE (2020)	10.4	63.4	75.3	10.4	61.2	66.4
+ UNT	13.7 $\uparrow$	65.0 $\uparrow$	75.8 $\uparrow$	12.7 $\uparrow$	63.4 $\uparrow$	66.8 $\uparrow$
NegSPL (2021b)	18.6	66.4	75.5	15.6	65.2	67.1
+ UNT	19.4 $\uparrow$	67.2 $\uparrow$	75.9 $\uparrow$	18.4 $\uparrow$	67.1 $\uparrow$	67.9 $\uparrow$

Table 6: Results of ablation study on different modules.

6.2 Ablations on Margin Softmax and Uncertainty Retraining

Table 6 (Top) shows an ablation study on the margin softmax regime, based on the data masking settings, where we study the impact of positive margin λ_+, negative margin λ_-, and Gaussian regularization respectively. According to the results, we find that the negative margin λ_- is the most effective, yet the effects of different components are complimentary. An ablation on the multiple triggers are shown in Appendix C.

In Table 6 (Bottom), we conduct an ablation study on our uncertainty-guided retraining mechanism and compare it to: (i) *w/o* uncertainty, which excludes the uncertainty interfering stage for learning, (ii) DirectPred, which retains the stage but uses predicted labels directly for model retraining, and (iii) BoostLearn, which considers half of the dataset to be clean and the other half to be unlabeled and conducts a bootstrapping process (Grézl and Karafiát, 2013). The results have ver-

Category	Example	Event Type
Negligence [51.1%]	1) ... less than 5,000 U.N troops could have stop the killings if Mr. Annan had ...	Die
	2) ... before the **genocide**, Major ... The ... informant that genocide was being ...	Attack
	3) The Justice party changed the constitution after taking power in the elections.	Elect
Light verbs [20.7%]	4) Anne-Marie got the couple 's 19-room home in New York state ...	TransferOwnership
	5) After he became SG [Secretary General], Annan commissioned a report ...	StartPosition
	6) ... GNP took two of the National Assembly seats; a splinter party got the third ...	StartPosition
Rare words [25.2%]	7) The troop opened its tank guns , opened its own mortars , decimated that unit ...	Attack
	8) ... Board would see it as leverage to seize power and pummel the office staff.	Attack
	9) Press speculation had ... while either divesting or inviting third parties to take ...	TransferOwnership
	10) But the general needed U.N. authorization to conduct such a raid and save lives.	Attack
Co-reference [3.0%]	11) ... in the 1994 genocide in Rwanda ... for not sending enough troops to stop it.	Attack

Table 7: Typical unlabeled cases in the development and test set of ACE 2005, grouped as four categories.

ified the effectiveness of our method — without the uncertainty mechanism (*w/o* uncertainty), the performance drops 2.4% in F1 on average for ACE and 2.1% for MAVEN. The major advantage of our method lies in that it can select reliable predictions for training — as evidence, we have checked the predictions with high probability (> 0.9) in the categorical distribution and found that 91.4% of them are correct. Finally, the results suggest that our uncertainty-guided mechanism can also promote OneIE and NegSPL, particularly in scenes with large unlabeled rates (e.g., $p = 10\%$ and $p = 50\%$).

6.3 Analysis of Unlabeled Cases

We explore common patterns of unlabeled events in Table 7. Indeed, 51.1% of them lack a discernible pattern, which could just be due to the annotator's negligence. For example, in case 2, the genocide event is labeled only in the first sentence but not in the subsequent one. The other patterns we find include light verbs (20.7%), such as got in case 4, rare words (25.2%), such as pummel in case 8, and co-reference based triggers (3%), such as it in case 11. These examples are hard for human annotators. We have also investigated the suspicious cases encountered in our re-annotation procedure. Aside from 11% merely mis-predicted by a model, we find two prevalent patterns: (i) compound nouns (54%), such as "election" in "create an election code", which does not refer to events, and (ii) definition violation (35%), such as "lobby" in "Bush plan to lobby allies ..." — though many event detectors predict "lobby" as a Meet event, but in the ACE event ontology, a Meet event is defined as "a meeting event is *physically located somewhere*". The comparison of cases of the control and challenge set is shown in Appendix A.2.

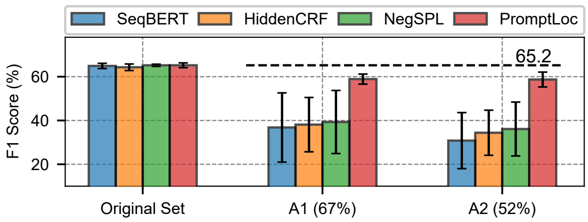

Figure 7: A real-world annotation test using the WikiEvents dataset as a case.

6.4 A Real-World Annotation Test

Finally, we conduct a real-world annotation test to investigate the practical applicability of our approach. Particularly, we use WikiEvents (Li et al., 2021a) as the test bed and employ two annotators to annotate events in 100 randomly selected training documents. For tractability, we only consider 10 most frequent event types and limit the annotation time to 4 hours. After deleting incorrect labels, we obtain A1 and A2, two sets with annotation rates of 67% and 52%, respectively. We then train models on A1, A2, and the original 100 labeled documents respectively and test them on the test set. The performances of different models are shown in Figure 8. According to the results, when trained on A1 and A2, previous models exhibit a significant drop in F1 (more than 25%). By comparison, our method achieves a good performance and performs comparably to methods that use the original training set for learning. This indicates its efficacy in dealing with the partial annotation issue.

7 Conclusion

In this study, we investigate the partial annotation problem in ED, a critical yet less-explored problem. We motivate a new learning model for ED and investigate its effectiveness in a variety of partial annotation settings. We also provide two re-annotated subsets of ACE 2005 to the community as a data contribution in order to establish a fair evaluation. In the future, we plan to investigate the theoretical aspects of our approach and increase its scope by applying it to other information extraction tasks suffering the partial annotation issue, such as named entity recognition and relation extraction.

8 Limitations

There are two limitations of this study that could be addressed in future research. First, this study focuses solely on the ED task. In the future, we seek to extend it to the overall event extraction (EE) task, which also includes the event argument extraction task, where a complete annotation is more challenging than in ED. Second, our study models the partially labeled training data instead of annotators. Indeed, the annotators produce the data, so building a model for annotators may be an essential way to address the partial learning problem. For example, an annotator may be more careless than others and generate more noisy data. Consequently, a robust model for the task should give a lower belief in the data of this annotator to improve learning. Lastly, our research raises no ethical issues because it focuses solely on the technical aspects of a normal information extraction problem.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No.62106016), the Open Projects Program of the State Key Laboratory of Multimodal Artificial Intelligence Systems, and the Tencent Open Fund.

References

David Ahn. 2006. The stages of event extraction. In *Proceedings of the Workshop on Annotating and Reasoning about Time and Events*, pages 1–8, Sydney, Australia. Association for Computational Linguistics.

Kedar Bellare and Andrew McCallum. 2007. Learning extractors from unlabeled text using relevant databases. In *AAAI*.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. 2020. A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1597–1607. PMLR.

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

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1267–1276, Brussels, Belgium. Association for Computational Linguistics.

S. Chopra, R. Hadsell, and Y. LeCun. 2005. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1.

Julie Medero Christopher Walker, Stephanie Strassel and Kazuaki Maeda. 2006. Ace 2005 multilingual training corpus ldc2006t06. In *Philadelphia: Linguistic Data Consortium*.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

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.

Charles Elkan and Keith Noto. 2008. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 213–220, New York, NY, USA. Association for Computing Machinery.

Xiaocheng Feng, Lifu Huang, Duyu Tang, Heng Ji, Bing Qin, and Ting Liu. 2016. A language-independent neural network for event detection. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 66–71, Berlin, Germany. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1050–1059, New York, New York, USA. PMLR.

Ralph Grishman. 1997. Information extraction: Techniques and challenges. In *Information Extraction A Multidisciplinary Approach to an Emerging Information Technology*, pages 10–27, Berlin, Heidelberg. Springer Berlin Heidelberg.

Frantiseli Grézl and Martin Karafiát. 2013. Semi-supervised bootstrapping approach for neural network feature extractor training. In *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 470–475.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Yu Hong, Jianfeng Zhang, Bin Ma, Jianmin Yao, Guodong Zhou, and Qiaoming Zhu. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1127–1136, Portland, Oregon, USA. Association for Computational Linguistics.

I-Hung Hsu, Kuan-Hao Huang, Elizabeth Boschee, Scott Miller, Prem Natarajan, Kai-Wei Chang, and Nanyun Peng. 2022. DEGREE: A data-efficient generation-based event extraction model. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1890–1908, Seattle, United States. Association for Computational Linguistics.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL-08: HLT*, pages 254–262, Columbus, Ohio. Association for Computational Linguistics.

Zhanming Jie, Pengjun Xie, Wei Lu, Ruixue Ding, and Linlin Li. 2019. Better modeling of incomplete annotations for named entity recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 729–734, Minneapolis, Minnesota. Association for Computational Linguistics.

Qi Li, Heng Ji, and Liang Huang. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 73–82, Sofia, Bulgaria. Association for Computational Linguistics.

Sha Li, Heng Ji, and Jiawei Han. 2021a. 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.

X. Li, Philip S. Yu, B. Liu, and S. Ng. 2009. Positive unlabeled learning for data stream classification. In *SDM*.

Yangming Li, Lemao Liu, and Shuming Shi. 2021b. Empirical analysis of unlabeled entity problem in named entity recognition. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Shasha Liao and Ralph Grishman. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 789–797, Uppsala, Sweden. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2018. Adaptive scaling for sparse detection in information extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1033–1043, Melbourne, Australia. 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.

B. Liu, Yang Dai, Xiaoli Li, Wee Sun Lee, and Philip S. Yu. 2003. Building text classifiers using positive and unlabeled examples. *Third IEEE International Conference on Data Mining*, pages 179–186.

Bing Liu, Wee Sun Lee, Philip S. Yu, and Xiaoli Li. 2002. Partially supervised classification of text documents. In *Machine Learning, Proceedings of the Nineteenth International Conference (ICML 2002), University of New South Wales, Sydney, Australia, July 8-12, 2002*, pages 387–394. Morgan Kaufmann.

Jian Liu, Yubo Chen, and Kang Liu. 2019a. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6754–6761.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020a. 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.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020b. How does context matter? on the robustness of event detection with context-selective mask generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2523–2532, Online. Association for Computational Linguistics.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2018a. Event detection via gated multilingual attention mechanism. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 4865–4872. AAAI Press.

Jian Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2019b. Neural cross-lingual event detection with minimal parallel resources. 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 738–748, Hong Kong, China. Association for Computational Linguistics.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022a. Multimedia event extraction from news with a unified contrastive learning framework. In *Proceedings of the 30th ACM International Conference on Multimedia*, MM '22, page 1945–1953, New York, NY, USA. Association for Computing Machinery.

Jian Liu, Yufeng Chen, and Jinan Xu. 2022b. Saliency as evidence: Event detection with trigger saliency attribution. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4573–4585, Dublin, Ireland. Association for Computational Linguistics.

Kun Liu, Yao Fu, Chuanqi Tan, Mosha Chen, Ningyu Zhang, Songfang Huang, and Sheng Gao. 2021. Noisy-labeled NER with confidence estimation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3437–3445, Online. Association for Computational Linguistics.

Shulin Liu, Yubo Chen, Kang Liu, and Jun Zhao. 2017. Exploiting argument information to improve event detection via supervised attention mechanisms. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1789–1798, Vancouver, Canada. Association for Computational Linguistics.

Xiao Liu, Heyan Huang, Ge Shi, and Bo Wang. 2022c. Dynamic prefix-tuning for generative template-based event extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5216–5228, Dublin, Ireland. Association for Computational Linguistics.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018b. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Zhengzhong Liu. 2018. *Diving Deep into Event Semantics*. Ph.D. thesis, Carnegie Mellon University.

Zhigang Liu, Wenzhong Shi, D. Li, and Qianqing Qin. 2005. Partially supervised classification - based on weighted unlabeled samples support vector machine. volume 3584, pages 118–129.

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.

Stephen Mayhew, Snigdha Chaturvedi, Chen-Tse Tsai, and Dan Roth. 2019. Named entity recognition with partially annotated training data. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 645–655, Hong Kong, China. Association for Computational Linguistics.

Minh Van Nguyen, Viet Dac Lai, and Thien Huu Nguyen. 2021. Cross-task instance representation interactions and label dependencies for joint information extraction with graph convolutional networks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 27–38, Online. Association for Computational Linguistics.

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

Thien Huu Nguyen and Ralph Grishman. 2015. Event detection and domain adaptation with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference* *on Natural Language Processing (Volume 2: Short Papers)*, pages 365–371, Beijing, China. Association for Computational Linguistics.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuanjing Huang. 2019. Distantly supervised named entity recognition using positive-unlabeled learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419, Florence, Italy. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Yuta Tsuboi, Hisashi Kashima, Shinsuke Mori, Hiroki Oda, and Yuji Matsumoto. 2008. Training conditional random fields using incomplete annotations. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 897–904, Manchester, UK. Coling 2008 Organizing Committee.

Sijia Wang, Mo Yu, Shiyu Chang, Lichao Sun, and Lifu Huang. 2022a. Query and extract: Refining event extraction as type-oriented binary decoding. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 169–182, Dublin, Ireland. Association for Computational Linguistics.

Sijia Wang, Mo Yu, and Lifu Huang. 2022b. The art of prompting: Event detection based on type specific prompts.

Xiaobo Wang, Shifeng Zhang, Zhen Lei, Si Liu, Xiaojie Guo, and Stan Z. Li. 2018. Ensemble soft-margin softmax loss for image classification. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, pages 992–998. ijcai.org.

Xiaozhi Wang, Xu Han, Zhiyuan Liu, Maosong Sun, and Peng Li. 2019. Adversarial training for weakly supervised event detection. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 998–1008, Minneapolis, Minnesota. 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.

Model	**Configuration**	**F1**
FFNNs	1 hidden layer; 100 hidden units	65.6
	1 hidden layer; 200 hidden units	66.1
	2 hidden layers; 100 hidden units each	66.3
	2 hidden layers; 200 hidden units each	65.5
CNNs	filter window of 2,3; 200 feature maps	68.7
	filter window of 2,3; 500 feature maps	68.6
	filter window of 2,3,4; 200 feature maps	68.7
	filter window of 2,3,4; 500 feature maps	67.0
RNNs	unidirectional; 100 hidden units	68.1
	unidirectional; 200 hidden units	67.9
	bidirectional; 100 hidden units	68.9
	bidirectional; 200 hidden units	68.0
GCNs	1 convolutional layer; 100 hidden units	70.0
	1 convolutional layer; 200 hidden units	69.7
	2 convolutional layers; 100 hidden units	68.8
	2 convolutional layers; 200 hidden units	69.1
BERT	$\text{Bert}_{\text{base}}$; cased tokenizer	71.8
	$\text{Bert}_{\text{base}}$; uncased tokenizer	70.1
	$\text{Bert}_{\text{large}}$; cased tokenizer	72.2
	$\text{Bert}_{\text{large}}$; uncased tokenizer	71.0

Table 8: Model details for potential unlabeled case identification, with their performances in the (original) ACE test set.

Fan Yang and Paul Vozila. 2014. Semi-supervised Chinese word segmentation using partial-label learning with conditional random fields. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–98, Doha, Qatar. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5284–5294, Florence, Italy. Association for Computational Linguistics.

A ACE 2005 Dataset Revision

Given that the ACE 2005 dataset is partially annotated and that comparing models on a partially annotated test set results in biased results, we revised the ACE 2005 development and test sets to create a fair benchmark. Specifically, we create an automatic method that incorporates (i) a potential false negative identification stage to identify all possible unlabeled cases and (ii) a human validation stage to manually validate each case.

A.1 Potential False Negative Identification

To identify potential unlabeled cases, we first train a set of 20 different ED models with diverse archi-

	Split	# Potential	# Validated	UL Rate
Dev. Set	Challenge	78	34 (43.6%)	6.7%
	Control	34	19 (55.9%)	3.8%
	Total	112	53 (47.3%)	10.5%
Test Set	Challenge	86	51 (59.3%)	12.0%
	Control	50	31 (62.0%)	7.3%
	Total	136	82 (60.2%)	19.3%

Table 9: Details of the revised ACE 2005 subsets. "UL Rate" is the ratio of unlabeled cases to labeled ones.

	ACE 2005			MAVEN		
Method	**10%**	**20%**	**30%**	**10%**	**20%**	**30%**
Hybrid† (2016)	7.4	22.3	37.1	6.4	17.7	31.7
OneIE (2020)	10.4	32.3	46.0	10.4	22.7	37.8
HiddenCRF (2019)	18.6	40.3	52.3	14.7	32.7	43.0
NegSPL (2021b)	20.6	42.3	54.3	15.6	35.7	46.0
No Prompting	60.7	68.2	72.5	51.8	59.9	62.0
Prompting *w* Type	61.3	68.9	72.7	52.3	60.3	62.3
Prompting *w* Desc.	62.0	68.7	71.9	52.1	60.5	62.5

Table 10: Results of different prompting strategies.

tectures[7] ranging from Feed-Forward Network Networks (Liu et al., 2017), Convolutional Network Networks (Chen et al., 2015), Recurrent Neural Networks (Nguyen et al., 2016), Graph Convolutional neural networks (Liu et al., 2018b) to pretrained language models (Yang et al., 2019), and then check their predictions on the development and test sets. The model details are shown in Table 8. Our intuition is that a wide range of ED models with various architectures can integrate a variety of inductive biases, and we regard any predicted trigger whose original label is O to be a potentially unlabeled example. Consequently, we uncover 112 and 136 potentially unlabeled cases on the ACE 2005 development and test sets respectively. To undertake a finer-grained analysis, we divide all the potential cases further into two groups: (i) a challenge set, in which more than half of the ED models predict an event label for a word whose initial label is O, and (ii) a control set in which fewer than half of the models do.

A.2 Human Validation

In the human validation stage, we manually check each potential case following the ACE Event Annotation Guidelines[8]. Specifically, we employ two annotators[9] to analyze each case independently, resulting in an inter-annotator agreement κ=0.81, and a third annotator to resolve the conflict. Table 9 summarizes the final human validation results — on the ACE 2005 development set, 53 unlabeled cases are finally confirmed (with a verification rate of 47.3%), producing a 10.5% percent ratio of unlabeled examples to labeled ones; on the ACE 2005 test set, 86 unlabeled cases are identified (with a verification rate of 60.2%), producing a 19.3 percent ratio of unlabeled examples to labeled ones. The high unlabeled ratio shows that the partial annotation problem is critical for the ACE 2005 corpus. Interestingly, we also note the challenge set has a lower validation rate than the control set. One reason for this is that the challenge set contains many spurious cases, such as compound nouns that are not event triggers, lowering the validation rate, whereas the control set contains many difficult cases, such as light verbs and unusual words that are ground-truth triggers missed by annotators, boosting the validation rate. We discuss the specific examples in Section 6.3.

[7]For each architecture, we create one model using the hyper-parameters specified in the original paper, as well as three variations with additional hidden layers.

[8]https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-events-guidelines-v5.4.3.pdf

[9]The annotators were recruited from a pool of students who had completed a high-level NLP class at the author's institution. Each annotator is compensated with around $70 for validating the total of 248 cases, resulting in an approximate payment of $0.28 per case. In § 6.3, a similar recruitment process and composition are adopted for annotations.

B Ablation on Prompting Strategy

We compare different prompting strategies, including "No Prompting", which does not uses prompting strategy, but build separate model for each event type. "Prompting *w* Type", which is our approach using event type as prompt. "Prompting *w* Description", which uses the event type description as the prompt. According to the results in Table 10, the prompting mechanism is not an important factor for improvement — the method without prompting (No Prompting) also yields good results. However, unlike the prompting method, which allows for natural parameter sharing, it necessitates the building of individual models for each event type, which may be costly in a real-world setting. Furthermore, we note that there is no noticeable difference when type or description are used as prompts.

C Performance with Multiple Triggers

We next investigate how well our approach performs in cases where the sentence contain multiple triggers. In the original ACE dataset, 25.1% (790

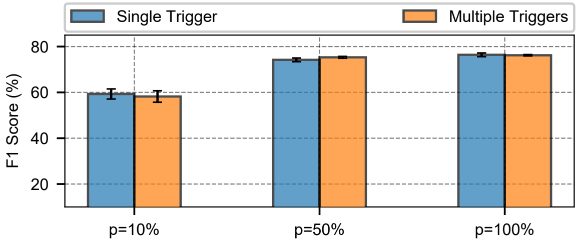

Figure 8: Ablations on multiple triggers on ACE.

out of 3136) of all sentences containing events have more than one event trigger. However, because our method treats different event types separately, it may only be impacted by sentences that contain two triggers of the same event type — such cases account for only 7% (245 out of 3136). Figure 8 shows the results of our approach for sentences with a single trigger and sentences with multiple triggers. The gap between the two is very small, indicating that our approach is effective for sentences with multiple triggers.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section 8

☑ A2. Did you discuss any potential risks of your work?
Section 8

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section 1

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Appendix A

☑ B1. Did you cite the creators of artifacts you used?
Appendix A, ACE 2005.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Appendix A

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Appendix A

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Appendix A

C ☑ Did you run computational experiments?

Section 5 and 6

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 4

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 4

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 5

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 4

D ☑ Did you use human annotators (e.g., crowdworkers) or research with human participants?
Appendix A

☑ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
Appendix A

☑ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
Appendix A

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
Not applicable. Left blank.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
Not applicable. Left blank.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
Not applicable. Left blank.