Unleashing Pre-trained Masked Language Model Knowledge for Label Signal Guided Event Detection

Mengnan Xiao, Ruifang He(✉), Junwei Zhang, Jinsong Ma, and Haodong Zhao

Tianjin Key Laboratory of Cognitive Computing and Application,
College of Intelligence and Computing, Tianjin University, Tianjin 300350, China
{mnxiao,rfhe,junwei,jsma,2021244138}@tju.edu.cn

Abstract. Event detection (ED) aims to recognize triggers and their types in sentences. Previous work employs distantly supervised methods or pre-trained language models to generate sentences containing events to alleviate data scarcity. Further, determining the spans and types of triggers is complex and may have deviations. In this paper, we propose to unleash Pre-trained Masked Language Model (PMLM) knowledge for label signal guided ED by a novel trigger augmentation. We directly generate triggers by leveraging the rich knowledge of PMLM through masking triggers. However, these newly replaced triggers may not correspond to the label of the masked trigger. To control such trigger augmentation noises, we design a label signal guided classification mechanism with event type-subtype guidance. To ensure the quality of generated triggers, a semantic consistency mechanism is introduced. Experimental results on the ACE2005 and FewEvent show the effectiveness of our proposed approach.

Keywords: Trigger augmentation · Label signal guided event classification · Sentence semantic consistency

1 Introduction

As a challenging subtask of event extraction, event detection (ED) aims to identify and classify triggers. As per the general ACE2005 annotation guideline: an event type contains one or more event subtypes. A sentence example is as follows: "*He lost an* ***election*** *to a dead man.*" Here, "***election***" triggers a "*Personnel: Elect*" event where "*Personnel*" is the event type and "*Elect*" is the event subtype.

So far, many methods have been proposed, extending from feature-based approaches to advanced deep learning methods [8,11]. Although previous methods achieve success in many aspects, data scarcity is a growing challenge that can not be ignored as mainstream models become bigger and bigger. The lack of training data seriously hinders the performance of existing methods, which are under the supervised learning paradigm and eager for the large training dataset. To alleviate the problem, Liu et al. [6] propose a multilingual approach

X. Wang et al. (Eds.): DASFAA 2023, LNCS 13945, pp. 572–581, 2023.
https://doi.org/10.1007/978-3-031-30675-4_42

by machine translation to bootstrap the source data. However, ensuring the mapping between tokens and labels across languages is complex and may have deviations. There also have been some efforts to enlarge training data for ED models by exploiting distantly supervised techniques [1,11,12]. Moreover, some work [8,13] leverages pre-trained language models to automatically generate training data for models. The common in these methods is to generate sentences containing events. However, there are two main weaknesses: 1) there are noises in the generated sentences and need extra mechanisms (such as knowledge distillation) to control; 2) ED is a token-level classification task, determining the spans and subtypes of triggers is difficult, and may have deviations.

To address the aforementioned problems, we explore directly generating proper triggers without changing the context, which can not only weaken noises but also reuse the labels of triggers in the original sentence. Inspired by Dai et al. [2], we propose a novel trigger augmentation approach by leveraging the existing pre-trained masked language model (PMLM) to automatically generate triggers. By replacing original triggers with generated ones, we can obtain candidate sentences with different triggers. Specially, we aim to fine-tune a PMLM on the existing training dataset by masking triggers so it can generate alternative triggers and corresponding scores. Yet trigger augmentation might still involve noises due to the complexity of natural language and the large vocabulary of PMLM. So we also design a **label signal guided classification mechanism** with **event type-subtype guidance**, including event type classification (ETC) and event subtype classification (ESC). The results of ETC serve as signals to guide ESC. Through the medium of ETC, we can calculate multiple times and finally select the maximum value of the product of ETC and ESC as the final result. In this manner, though the result of ETC is not correct, the final result may also be right. We also design a **sentence semantic consistency mechanism** that makes the semantics between the candidate and original sentence as similar as possible to ensure the quality of the generated triggers. With the right generated triggers, the semantics of sentences are naturally similar. Our contributions in this paper can be summarized as follows:

- Propose a novel trigger augmentation approach (called PMLMLS) for ED to directly generate alternative triggers by leveraging the knowledge of PMLM;
- Build a label signal guided classification mechanism with event type-subtype guidance for ED which helps control noises in trigger augmentation;
- Employ a sentence semantic consistency mechanism to ensure the quality of generated triggers;
- Experimental results on the ACE2005 and FewEvent demonstrate the effectiveness of our method and achieve state-of-the-art performance.

2 Methodology

Figure 1 shows the proposed PMLMLS model, which leverages the knowledge of the pre-trained masked language model (PMLM) to improve ED. The model consists of two stages: (1) **Trigger Augmentation**: to employ PMLM to generate

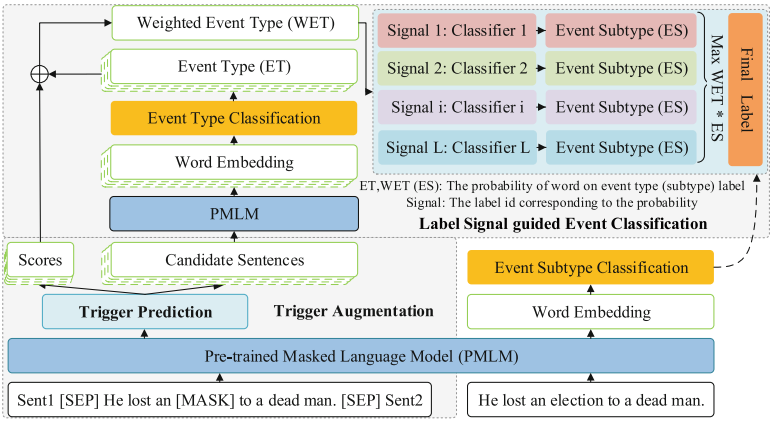

Fig. 1. The overview of our proposed PMLMLS.

alternative triggers and corresponding scores; (2) **Label Signal Guided Event Classification**: to utilize label signal to guide event type-subtype classification which helps control noises in (1).

2.1 Trigger Augmentation

As presented in Sec. 1, our motivation is to obtain proper candidate triggers without changing the context. The overall strategy is to mask the trigger with a special token and leverage PMLM to generate the candidates. Formally, assume that $\boldsymbol{x} = [x_1, \ldots, x_i, \ldots, x_n]$ is a sentence of n tokens with only one trigger located at x_i, the masked sentence $\boldsymbol{x'}$ would have the form: $\boldsymbol{x'} = [x_1, \ldots, [MASK], \ldots, x_n]$ where $[MASK]$ is the special token to symbolize the trigger. $\boldsymbol{x'}$ is then employed as the input of PMLM to obtain the representation $\boldsymbol{h}_{\text{mask}}$ of $[MASK]$:

$$\boldsymbol{h}_{\text{mask}} = \text{PMLM}(\boldsymbol{x'}) \in R^d \tag{1}$$

where d denotes the dimension of the hidden layer in PMLM. Then we utilize PMLM head (i.e., LMhead) to obtain top k triggers $\boldsymbol{T} = [t_1, \ldots, t_i, \ldots, t_k]$ and corresponding scores $\boldsymbol{s} = [s_1, \ldots, s_i, \ldots, s_k]$:

$$(\boldsymbol{T}, \boldsymbol{s}) = \text{LMhead}(\boldsymbol{h}_{\text{mask}}) \tag{2}$$

where LMhead is a pre-trained two-layer non-linear classifier with layer normalization and the output dimension is the size of the vocabulary of PMLM. The score s_i is the probability of LMhead on the corresponding candidate trigger t_i. Note that the sum of $\boldsymbol{s}$ is not equal to 1 and then we normalize $\boldsymbol{s}$:

$$s_i = \frac{s_i}{\sum_{j=1}^{k} s_j} \in R \tag{3}$$

Before we fill $\boldsymbol{T}$ into $[MASK]$ and obtain k candidate sentences, we preliminarily judge the quality of $\boldsymbol{T}$ through $x_i \in \boldsymbol{T}$ or not. If $x_i \notin \boldsymbol{T}$, then the quality of $\boldsymbol{T}$ is unreliable and we will abandon it.

Considering that the trigger is usually the core word (verb or noun) of the sentence, there would be many choices in the scope of the vocabulary of PMLM. Sometimes it even generates candidates that are appropriate in the context but completely irrelevant to the original word with high scores (e.g. the example in the introduction). To help PMLM generate suitable candidates that are related to the original trigger, we add the previous and next sentences of $\boldsymbol{x}$ as a prompt to $\boldsymbol{x'}$. The enriched $\boldsymbol{x'}$ would have the form: $\boldsymbol{x'} = \left[\boldsymbol{Sent1}, [SEP], x_1, \ldots, [MASK], \ldots, x_n, [SEP], \boldsymbol{Sent2}\right]$ where $[SEP]$ is the special token to identify the span of sentences.

2.2 Label Signal Guided Event Classification

To control noises in trigger augmentation, we design a **label signal guided classification mechanism** with **event type-subtype guidance**.

Label Signal Guided Classification Mechanism: Considering that an event type consists of one or more event subtypes, we design a label signal guided classification mechanism, first event type classification (ETC) then event subtype classification (ESC). Formally, as per the pre-defined event schema, we have an event type set $\mathcal{C}$ and an event subtype set $\mathcal{Y}$. The overall goal is to predict all events in gold set $\mathcal{E}_{\boldsymbol{x}}$ of the sentence $\boldsymbol{x}$. We aim to maximize the joint likelihood of training data $\mathcal{D}$:

$$\prod_{\boldsymbol{x}\in\mathcal{D}}\left[\prod_{(t,c,y)\in\mathcal{E}_{\boldsymbol{x}}} p\big((t,c,y)\mid \boldsymbol{x}\big)\right] = \prod_{\boldsymbol{x}\in\mathcal{D}}\left[\prod_{t\in\mathcal{T}_{\boldsymbol{x}}}\Big[p(t\mid \boldsymbol{x})p(c\mid \boldsymbol{x},t)p(y\mid \boldsymbol{x},t,c)\Big]\right] \tag{4}$$

where $\mathcal{T}_{\boldsymbol{x}}$ denotes the triggers set occurring in $\boldsymbol{x}$, t denotes the trigger in $\mathcal{T}_{\boldsymbol{x}}$, c denotes the event type of t, and y denotes the event subtype of t. The result of ETC is leveraged as a signal to guide ESC. It is a tree with a layer height of 3, the root node is the trigger, and the second and third layers are event types and subtypes respectively. The children of the second layer node are the event subtypes contained in the event type, and the weights of edges are probabilities of ETC and ESC classifiers. When classification, the trigger selects a path to the leaf node in a depth-first search (DFS) based on the edge weight.

To control noises in the trigger augmentation, we do not only utilize the label corresponding to the maximum value of the ETC prediction result as a signal but the top m results as signals. When starting from each node, instead of choosing one path, we choose m paths as per the signals. Finally, the maximum value of the product of all edge weights on the search path is employed as the final result. In this manner, though the result of ETC is not correct, the final result may also be right. We can obtain the global optimal solution to a certain extent through multiple searches.

Event Type-Subtype Guidance Classification Network: As per the aforementioned mechanism, we build an event type-subtype guidance classification network containing ETC and ESC. The thought of ETC and ESC are similar while ETC is trained on candidate sentences and obtain event type results, ESC is trained on original sentences and obtain event subtype results as per the results of ETC. Assume that $\hat{\boldsymbol{X}}$ is the candidate sentences obtained by the original sentence $\boldsymbol{x}$ after Sec. 2.1. Then we utilize the PMLM to obtain the hidden presentation of tokens in $\hat{\boldsymbol{X}}$ and $\boldsymbol{x}$:

$$\hat{\boldsymbol{H}} = \text{PMLM}(\hat{\boldsymbol{X}}) \qquad \boldsymbol{H} = \text{PMLM}(\boldsymbol{x}) \tag{5}$$

where the PMLM is the one in Sec. 2.1, they share weights, $\hat{\boldsymbol{H}}$ is the embedding of tokens in candidate sentences, and $\boldsymbol{H}$ is the embedding of tokens in the original sentence. Then $\hat{\boldsymbol{H}}$ is used as the input of ETC to obtain the event type result $\hat{\boldsymbol{C}}$:

$$\hat{\boldsymbol{C}} = \text{ETC}(\hat{\boldsymbol{H}}) \tag{6}$$

where ETC is a two-layer non-linear classifier with dropout and layer normalization. In addition, we obtain the score of candidate sentence $\boldsymbol{s}$ by Eq. 2 and 3. Therefore we obtain the weighted probability over event type $\hat{\boldsymbol{p}}$ by the product of $\hat{\boldsymbol{C}}$ and $\boldsymbol{s}$ normalized by softmax$(\cdot)$:

$$\hat{\boldsymbol{p}} = \text{softmax}\Big(\sum_{i=1}^{z} s_i \hat{\boldsymbol{C}}_i\Big) \tag{7}$$

Then the top m probability $\boldsymbol{v}$ and the corresponding event type label id $\boldsymbol{\ell}$ of $\hat{\boldsymbol{p}}$ consist of signals to guide ESC:

$$\boldsymbol{y} = \max\{v_i \cdot \text{softmax}(\text{ESC}_{\ell_i}(\boldsymbol{H}))|i = 1, \ldots, m\} \tag{8}$$

where ESC contains $\boldsymbol{L}$ classifiers and each is a two-layer non-linear classifier with dropout and layer normalization. $\boldsymbol{L}$ denotes the number of event types, ESC_{ℓ_i} denotes choosing the ℓ_i-th classifier as per ℓ_i, $v_i \cdot \text{softmax}(\text{ESC}_{\ell_i}(\boldsymbol{H}))$ denotes the product of probabilities, and $\boldsymbol{y}$ denotes the final event subtype result of tokens in $\boldsymbol{x}$.

2.3 Training

This section describes the training of our model. In addition, to further make sure the quality of generated triggers, sentence semantic consistency is introduced.

Sentence Semantic Consistency: In Sec. 2.1, we preliminarily judge the quality of the candidate triggers by $x_i \in \boldsymbol{T}$ or not. But for $x \in \boldsymbol{T} \setminus \{x_i\}$, the quality can not be guaranteed. Considering the only difference between candidate and original sentences is triggers. Therefore, we try to make the semantics between the candidate and the original sentence as similar as possible. In this

work, we utilize the mean squared error between $\hat{\boldsymbol{H}}_{\text{cls}}$ and $\boldsymbol{H}_{\text{cls}}$ as a supervised target for the loss function:

$$\mathcal{L}_s = \frac{1}{|\boldsymbol{H}_{\text{cls}}|} \sum_{i=1}^{|\boldsymbol{H}_{\text{cls}}|} (\boldsymbol{H}_{\text{cls},i} - \hat{\boldsymbol{H}}_{\text{cls},i})^2 \tag{9}$$

where $\hat{\boldsymbol{H}}_{\text{cls}}$ and $\boldsymbol{H}_{\text{cls}}$ denote the semantics of candidate and original sentences respectively, $|\boldsymbol{H}_{\text{cls}}|$ denotes the dimension of $\boldsymbol{H}_{\text{cls}}$, and $\boldsymbol{H}_{\text{cls},i}$ denotes the i-th element of $\boldsymbol{H}_{\text{cls}}$.

Joint training: Finally, to train PMLMLS, the following combined loss function is employed:

$$\mathcal{L} = \mathcal{L}_{\text{ETC}} + \alpha \mathcal{L}_{\text{ESC}} + \beta \mathcal{L}_s \tag{10}$$

where $\mathcal{L}_{\text{ETC}}$ employs cross-entropy loss between the real and predicted event type labels, $\mathcal{L}_{\text{ESC}}$ employs the same loss on the real and predicted event subtype labels, α and β are the trade-off parameters.

3 Experiments

In this section, we explore the following questions:

Q1: Can PMLMLS better utilize the knowledge of PMLM to boost the performance of ED? Q2: Is every module essential? Q3: How do hyper-parameters affect the performance of PMLMLS?

3.1 Settings

Datasets: We conduct experiments on the event detection benchmark ACE2005, which has 599 English annotated documents and 8 event types total of 33 event subtypes. The same split as the previous work [8,11] is used.

In addition, we also conduct experiments on another benchmark FewEvent [3], which contains 70,852 instances for 19 event types graded into 100 event subtypes in total. To validate the performance of PMLMLS in the data scarcity scenario, we randomly select 30 instances for each event subtype in each trial. In a trial, the proportion of instances for each event subtype in the training, development, and test set are 70%, 10%, and 20% respectively.

For evaluation, we employ standard Precision (P), Recall (R), and the F_1 score following the previous work [8,11]. And we employ the average of 5 experimental results as the final result.

Baselines: To verify PMLMLS, we compare our method with models based on the aforementioned two strategies and other SOTA methods.

For ACE2005, we compare PMLMLS with several state-of-the-art models in three categories: (1) Multi-label classification model: **DMCNN** [1], **MLBiNet** [7], and **ED3C** [9]; (2) QA-based model: **RCEE_ER** [5]; (3) Data augmentation model: **GMLATT** [6], **DMBERT** [12], **DRMM** [10], **EKD** [11], and **GPTEDOT** [8]. For FewEvent, we compare PMLMLS with the following models: **PLMEE** [13], **DMBERT** [12], and **EEQA** [4].

Table 1. Overall performance (a) and ablation study (b) on the ACE2005 test set. In (a), $*$ indicates models based on PLMs. In (b), all the models in this table utilize RoBERTa-base. (The same as below)

Model	P	R	F_1
DMCNN [1]	79.7	69.6	74.3
GMLATT [6]	78.9	66.9	72.4
DMBERT* [12]	77.6	71.8	74.6
RCEE_ER* [5]	75.6	74.2	74.9
DRMM* [10]	77.9	74.8	76.3
EKD* [11]	79.1	78.0	78.5
MLBiNet [7]	74.7	83.0	78.6
ED3C* [9]	75.1	83.5	79.1
GPTEDOT* [8]	82.3	76.3	79.2
PMLMLS (ours)*	76.6	82.8	**79.6**

(a) Overall performance

Model	P	R	F_1
ED	74.3	73.0	73.6
LSED	74.8	75.2	75.0
PMLMED^{-all}	73.4	79.0	76.1
PMLMED^{-cp}	76.2	78.0	77.1
PMLMED^{-ssc}	75.8	78.9	77.3
PMLMED	76.0	80.7	78.3
PMLMLS^{-all}	74.0	80.2	77.0
PMLMLS^{-cp}	76.8	80.5	78.6
PMLMLS^{-ssc}	76.6	80.5	78.5
PMLMLS	76.6	**82.8**	**79.6**

(b) Ablation study

Implementations: We choose RoBERTa-base as the pre-trained masked language model and experiment with MindSpore. The hidden state and dropout of ETC and ESC are set to 768 and 0.1 respectively. The trade-off parameters α and β are set to 0.6 and 0.2 respectively. The learning rate is set to 1e−5 for the Adam optimizer and the batch size of 4 is employed during training. k is set to 4 denotes trigger augmentation will generate 4 alternative triggers. m is set to 2 denotes ESC will compute 2 times as per the top 2 probability of ETC. The epoch is set to 50 and the early stop is set to 8.

3.2 Overall Performance

Table 1 (a) presents the performance of all baselines and PMLMLS on the ACE2005 test set. For *Q1*, we can observe that:

1) By fully leveraging the rich knowledge of the pre-trained masked language model and label signal guided classification, PMLMLS outperforms all baselines with simpler architecture. Our method, only using a shared PMLM, surpasses GPTEDOT [8] which utilizes two PLMs and achieves competitive performance with the new SOTA. Furthermore, compared with other models that need the extra complicated module to control noise (e.g. knowledge distillation), PMLMLS only utilizes a two-stage classification based on label signal.

2) By directly generating alternative triggers from the pre-trained masked language model, PMLMLS achieves better results compared to other data argumentation models. Our method improves F_1 by 1.0% and 0.4% over the SOTA EKD [11] based on distant supervision and GPTEDOT [8] based on GPT-2 respectively. Compared with generating sentences containing events, directly generating alternative triggers can weaken noise and reuse the label of the original sentence.

Table 2. Overall performance and ablation study on the FewEvent test set.

Model	P	R	F_1
PLMEE* [13]	60.1	58.2	59.1
DMBERT* [12]	60.3	58.4	59.3
EEQA* [4]	61.2	59.3	60.2
PMLMLS (ours)*	**62.0**	**60.3**	**61.1**

(a) Overall performance

Model	P	R	F_1
ED	60.2	53.3	56.5
LSED	60.7	54.1	57.2
PMLMED	57.4	59.6	58.5
PMLMLS	**62.0**	**60.3**	**61.1**

(b) Ablation study

Table 2 (a) presents the performance of PMLMLS on the FewEvent test set. We can see that: our proposed model has an improvement compared with all baselines, thus further confirming the advantages of PMLMLS for ED.

3.3 Ablation Study

To verify *Q2*, for ACE2005, first, for the importance of label signal, we take the following baselines: (1) ED: the base model based on the PMLM without trigger augmentation and label signal guided classification; (2) LSED: based on (1), LSED adds label signal guided classification. Second, based on the trigger augmentation, three components need to be evaluated, the previous and next sentences prompt (context prompt, cp), label signal guided classification (ls), and sentence semantic consistency (ssc) respectively. There are a total of 8 combinations, one of which is PMLMLS. Therefore, we choose the remaining 7 combinations as degradation experiments. They are (3) PMLMED$^{\text{-all}}$: the baseline model based on trigger augmentation, without cp, ls, and ssc; (4) PMLMED$^{\text{-cp}}$: based on (3), add ssc; (5) PMLMED$^{\text{-ssc}}$: based on (3), add cp; (6) PMLMED: based on (3), add cp and ssc; (7) PMLMLS$^{\text{-all}}$: the baseline model based on trigger augmentation and label signal guided classification, without cp and ssc; (8) PMLMLS$^{\text{-cp}}$: based on (7), add ssc; (9) PMLMLS$^{\text{-ssc}}$: based on (7), add cp.

For FewEvent, there is no concept of the document, and the training data is in the form of sentences, so there is no context prompt. Degradation experiments include: (1) ED: the baseline only utilizes RoBERTa-base; (2) LSED: based on (1), add label signal guided classification; (3) PMLMED, based on (1), add trigger augmentation. From Table 1 (b), we can observe that:

1) The trigger augmentation, cp, ssc, and ls are necessary for PMLMLS to achieve the highest performance. Remove any component, performance will decrease. In particular, the F_1 score decreases by 1.0%, 1.1%, 1.3%, and 4.6% when removing cp, ssc, ls, and trigger augmentation. Note that when removing trigger augmentation, cp and ssc will also remove.

2) Label signal guided classification is helpful at any time. There are 10 degradation experiments, and we can divide them into 5 groups: a) ED and LSED; b) PMLMED$^{\text{-all}}$ and PMLMLS$^{\text{-all}}$; c) PMLMED$^{\text{-cp}}$ and PMLMLS$^{\text{-cp}}$; d) PMLMED$^{\text{-ssc}}$ and PMLMLS$^{\text{-ssc}}$; e) PMLMED and PMLMLS. The difference

Table 3. Performance of PMLMLS on the ACE2005 test set with different k and m.

	k						m		
	1	2	3	4	5	6	1	2	3
P	74.6	75.5	76.9	76.6	74.9	75.6	75.8	76.6	76.8
R	75.0	78.4	79.5	**82.7**	80.9	74.5	81.6	82.7	82.9
F_1	74.8	76.9	78.2	**79.6**	77.8	75.1	78.6	79.6	79.7

between the two experiments in each group is whether to perform label signal guided classification. We can see that the effect of using label signal guided classification in each set of experiments is better than not using and the average improvement is 1.3%.

3) Adding additional training data is an effective method for data scarcity. Yet it will inevitably introduce noises. The key is to control noises while increasing the training data. Compared with ED, PMLMED$^{\text{-all}}$ adds additional training data without extra mechanisms to control noises, we can see that the F_1 score increases, but at the cost of a decrease in P. When additional mechanisms (cp, ssc, or both) are added to control noise, the scores of P, R, and F_1 increase over ED. In addition, from Table 2 (b), we can see that: Compared with ACE2005, the effect of each module is better in the scarcer FewEvent.

3.4 Parameter Analysis

To illustrate *Q3*, in addition to the hyperparameters of the neural network, two additional hyperparameters need to be set. They are the number of alternative triggers generated for the masked trigger k and the top m results of ETC consist of signals to guide ESC.

To study the importance of k, we experiment with different k on the ACE2005. From the left of Table 3, the highest performance of the proposed model is achieved when k is 4 which denotes trigger augmentation generates 4 alternative triggers for the masked trigger. More specially, when $k \leq 3$, as k increases, P, R, and F_1 increase. We can see the knowledge of the pre-trained masked language model can predict proper and various triggers, alleviate data scarcity and improve performance. When k equals 4, P drops slightly compared to k equals 3. Though achieving the highest, we can see it is a bit noisy but more profitable. When $k \geq 5$, noise dominates and affects the performance of the ED model.

To provide more insights into the influence of label signal guided classification, we conduct experiments with different m on the ACE2005. From the right of Table 3, we can see that with the increment of m, the performance of PMLMLS improves. That is because PMLMLS makes multiple judgments when making the final result, weakening the interference of noise. Note that using label signal guided classification will affect the parallelism and need more time since we need to select the corresponding classifier in ESC as per the results of ETC. Even though the F_1 score when $m = 3$ is higher than when $m = 2$, however, the improvement is slight. So we select $m = 2$ as the final result to balance F_1 and time costing.

4 Conclusions

In this paper, we propose a novel trigger augmentation method (called PMLMLS) for ED leveraging the rich knowledge of the pre-trained masked language model. Unlike other data augmentation methods that generate sentences containing events, PMLMLS directly generates alternative triggers by masking triggers to weaken noises from the source. We also design a label signal guided classification mechanism with event type-subtype guidance to alleviate the noises in trigger augmentation. Sentence semantic consistency is also introduced to ensure the quality of generated triggers. Comprehensive experimental results on the ACE2005 and FewEvent demonstrate the effectiveness of the proposed method.

Acknowledgments. Our work is supported by the National Natural Science Foundation of China (61976154) and CAAI-Huawei MindSpore Open Fund.

References

1. Chen, Y., Liu, S., Zhang, X., Liu, K., Zhao, J.: Automatically labeled data generation for large scale event extraction. In: ACL, pp. 409–419 (2017)
2. Dai, H., Song, Y., Wang, H.: Ultra-fine entity typing with weak supervision from a masked language model. In: ACL, pp. 1790–1799 (2021)
3. Deng, S., Zhang, N., Kang, J., Zhang, Y., Zhang, W., Chen, H.: Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In: WSDM, pp. 151–159 (2020)
4. Du, X., Cardie, C.: Event extraction by answering (almost) natural questions. In: EMNLP, pp. 671–683 (2020)
5. Liu, J., Chen, Y., Liu, K., Bi, W., Liu, X.: Event extraction as machine reading comprehension. In: EMNLP, pp. 1641–1651 (2020)
6. Liu, J., Chen, Y., Liu, K., Zhao, J.: Event detection via gated multilingual attention mechanism. In: AAAI, pp. 4865–4872 (2018)
7. Lou, D., Liao, Z., Deng, S., Zhang, N., Chen, H.: MLBiNet: A cross-sentence collective event detection network. In: ACL, pp. 4829–4839 (2021)
8. Pouran, Ben Veyseh, A., Lai, V., Dernoncourt, F., Nguyen, T.H.: unleash GPT-2 power for event detection. In: ACL, pp. 6271–6282 (2021)
9. Veyseh, P.B.A., Nguyen, M.V., Ngo Trung, N., Min, B., Nguyen, T.H.: Modeling document-level context for event detection via important context selection. In: EMNLP, pp. 5403–5413 (2021)
10. Tong, M., et al.: Image enhanced event detection in news articles. In: AAAI, pp. 9040–9047 (2020)
11. Tong, M., et al.: Improving event detection via open-domain trigger knowledge. In: ACL, pp. 5887–5897 (2020)
12. Wang, X., Han, X., Liu, Z., Sun, M., Li, P.: Adversarial training for weakly supervised event detection. In: NAACL:HLT, pp. 998–1008 (2019)
13. Yang, S., Feng, D., Qiao, L., Kan, Z., Li, D.: Exploring pre-trained language models for event extraction and generation. In: ACL, pp. 5284–5294 (2019)