Prompt-Enhanced Prototype Framework for Few-shot Event Detection

Xu Liu
Beijing University of Posts and Telecommunications
Beijing, China
pandaxu@bupt.edu.cn

Xinming Chen
Beijing University of Posts and Telecommunications
Beijing, China
cxm@bupt.edu.cn

Yangfu Zhu
Beijing University of Posts and Telecommunications
Beijing, China
zhuyangfu@bupt.edu.cn

Bin Wu*
Beijing University of Posts and Telecommunications
Beijing, China
*corresponding author
wubin@bupt.edu.cn

2024 International Joint Conference on Neural Networks (IJCNN) | 979-8-3503-5931-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IJCNN60899.2024.10651359

***Abstract*—Few-shot event detection (ED) aims at identifying and typing event mentions from text with limited annotations. Most existing methods for few-shot ED use event ontology and related knowledge to construct prototypes and fail to fully leverage the rich knowledge of pre-trained language models (PLMs) which could help improve the representation of prototypes. Motivated by this, we propose an prompt-enhanced prototype framework which combines prototype and prompt for few-shot ED. Considering the scarcity of labeled data, we also introduce contrastive learning to enrich prototypes. Specifically, we use heuristic rules to align FrameNet with annotated data to get corresponding prompts for each event and convert them into prompt prototype. We then leverage contrastive learning to aggregate event mentions into prototypes and maintain these prototypes for few-shot ED. Furthermore, We explore diverse prompt formats for representing prompt prototypes and introduce a more comprehensive lexical prompt which improves the performance of few-shot ED. We conduct extensive experiments on the MAVEN corpus to reveal the effectiveness of the proposed framework compared to state-of-the-art methods.**

I. Introduction

Event Detection (ED) constitutes a vital subtask within event extraction, focusing on the identification and categorization of event mentions in natural language text. For instance, in *"Despite this, two Swedish ships were **burned** and soon sank"*, the trigger word is *burned*, indicating an "**Destroying**" event. Supervised approaches, particularly deep neural networks [1], [2], have exhibited impressive performance, but they heavily rely on a large number of manual annotations. However, in real-world applications, obtaining enough labeled data is often impractical. Therefore, methods that demonstrate effective generalization with limited labeled samples are highly desirable for event detection, i.e., few-shot ED.

Numerous studies have extensively explored few-shot ED. ProtoNet [3] and its variants[4], [5], [6], [7] construct prototypes using a subset of sample mentions for few-shot ED. However, these methods suffer from the problem of inaccurate prototype representation. To tackle this problem, a series of works leverage related knowledge to enhance the representation of prototypes, including AMR graphs [8], definitions [9], FrameNet [10] and event-event relations [11]. Nevertheless, these methods encounter challenges related to insufficient learning of discriminative representations in few-shot settings. Recently, several studies [12], [13], [14] have introduced Contrastive Learning to address this challenge. However, all these methods fail to fail to fully leverage the rich knowledge of PLMs which could help improve the representation of prototypes. With the development of pre-trained language models, prompt learning [15], [16], [17], [18], [19] is applied to few-shot ED. Although these methods achieve encouraging progress through prompt learning, the performance remains unsatisfactory [14] compared to the prototype methods owing to the inherent disparity between ED and pre-training objective of language models.

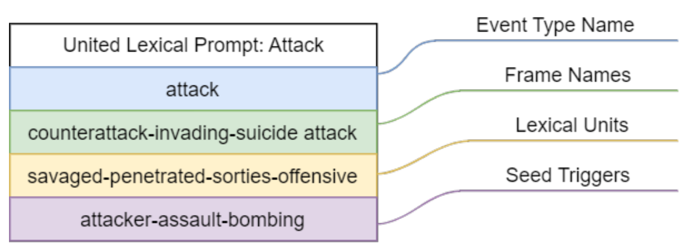

Fig. 1. Example of *United Lexical Prompt* for event type **Attack**

To develop an effective model for few-shot ED, in this paper, we propose an prompt-enhanced prototype framework. In which, the prototype-based model serves as the backbone, and prompts are used to enhance the prototype representation. More specifically, We assign two branches for each event type, one of which uses the rich knowledge of PLMs via prompts to improve the representation of the prototype, and the other uses contrastive learning to aggregate event mentions into prototypes. Besides, we explore diverse prompt formats for representing event prototypes, propose a more comprehensive description *United Lexical Prompt* and prove that describing event prototypes with comprehensive lexical prompts improves the performance of ED with limited labeled samples. *United Lexical Prompt* can be automatically extracted by aligning FrameNet to the event ontology using heuristic rules. Figure 1 shows an example of *United Lexical Prompt* which contains essential lexical information about event type **Attack**. We conducted experimental studies on MAVEN [20] corpus to demonstrate its benefits in few-shot event detection. Our contributions are summarized as follows:

- We developed an prompt-enhanced prototype framework which combines prototype and prompt for few-shot ED.
- We explore diverse prompt formats for representing event prototypes and prove that describing event prototypes with comprehensive lexical prompts (*United Lexical Prompt*) improves the performance of few-shot ED.
- We achieve the best performance (F1 score) on MAVEN corpus, surpassing the state-of-the-art by 1.0 points.

II. Related Work

While event detection has made significant progress within the supervised training paradigm [1], [2], [21], [22], [23], [24], its effective generalization to new languages, domains, or types faces challenges, particularly in cases where annotations are insufficient. Consequently, there has been a significant amount of research dedicated to few-shot event detection in recent years. Following previous work [14], the current few-shot event detection methods can be classified into two categories: prototype-based and prompt-based methods.

Prototype-based methods maintain one or more prototypes, each functioning as an embedding representing a specific event type. These approaches ascertain the event type of each event mention based on the proximity of their representations to the respective prototypes. In addressing the challenge of constructing prototypes, several methods [3], [4], [5], [6] have been proposed to utilize a subset of sample mentions. Furthermore, a series of works leverage related knowledge to improve the representation of prototypes, including AMR graphs [8], definitions [9], FrameNet [10] and event-event relations [11]. Recently, several studies [13], [14] have introduced Contrastive Learning to improve the representations of prototypes. However, it remains uncertain whether these are the optimal representations of prototypes.

Prompt-based methods aim to fully utilise the capabilities of PLMs by transforming the event detection task into a more familiar pre-training task for the PLMs, e.g., machine reading comprehension [15], [16], natural language inference [25], conditional generation [26], [18], and the cloze task [19]. However, creating meaningful prompts for event detection with exceedingly scarce annotations remains a challenge owing to the inherent disparity between ED and pre-training objective of language models. In this work, we explore different formats of prompts, including template-based and FrameNet-based, to construct prototypes for few-shot event detection task.

III. Preliminary

This work aims to compare different prompt formats for representing prototype in the context of few-shot event detection. Firstly, we present a formulation for event detection (ED). Subsequently, we elaborate on various forms of prompts designed to convey event prototype.

A. Problem Formulation

In the literature, ED is commonly defined as either a span classification task [27] or a sequence labeling [28] task. In this paper, we employ span classification paradigm as MAVEN offers official spans for candidate triggers (encompassing negative samples). Here we provide a brief overview of the span classification paradigm.

Given a corpus D and its annotated event types T, we select an sentence $S = [w_1, w_2, ..., w_n]$(with n words) along with candidate triggers $C = [c_1, c_2, ..., c_m]$(with m spans) which dicates spans in sentence S. ED aims to assign a label $y_i \in Y$ to each span $c_i \in C$ so the label sequence $Y = (T \cup N)$ can capture event triggers and types in S. Here, we also include a special type N to indicate that a word is not an event trigger or beyonds pre-defined types T.

B. Event Type Prompts

We compare seven distinct prompt formats for representing event prototype.

1) Event Type Name: Type name stands out as the most direct and intuitive representation of an event prototype. We follow the previous specific research about prompts on event type name [29] and incorporate them into all subsequent text-based event prototype prompts owing to their fundamental and easily distinguishable nature.

2) Frame Names: Type name might not fully encapsulate the semantics of an event type, given the inherent ambiguity of type names and the diversity of event mentions. The frame names derived from FrameNet [30] instead describe the meaning of the event prototype more comprehensively, which usually consists of one to four phrases. For example, *sending* can either refer to *product delivery* or *commerce money transfer* or *Sending* which can be found in frame names.

3) Frame Definition: In certain instances, frame names may consist of only one or two names, providing an incomplete representation of the event prototype. Intuitively, Frame definitions contribute to enhancing the semantic information of events. As an illustrative example, let's consider the event type $Killing$ from MAVEN. Its frame definition is *A Killer or Cause causes the death of the Victim.*

4) Lexical Units: lexical units (LUs) refer to vocabulary items that can evoke a specific frame. For example, $Applyheat$ serves as a frame, and the associated LUs include words such as *bake blanch boil broil brown simmer steam.* These lexical units possess specific semantic associations, facilitating the description of specific types of events. For each word in lexical units, we select the top-K^4 ranked words for representing event prototype based on the cosine similarity to event type name.

5) Prototype Seed Triggers: The representation based on seed triggers comprises the type name and a list of prototype triggers. Following previous research [31], given an event type t and its annotated triggers, we calculate the frequency f_o of each trigger word across the entire training dataset and its frequency f_t of being labeled as an event trigger of type t. The top-K^4 words are chosen as prototype triggers, determined by the probability ratio f_t/f_o for each word. Thus, for the event type Know, we represent it as *know witnessed understood delineated well_known*.

6) Apex Prompt: The prior study [29] introduces a comprehensive description, denoted as APEX prompt, for each event type. This prompt is formed by concatenating the event type name, seed triggers, and the definition, encompassing all argument roles, e.g., *killing, extermination slaughtered murderers manslaughte, A Killer or Cause causes the death of the Victim*. Since the MAVEN corpus does not give an official definition of the event type, we choose the frame definition as event definition.

7) United Lexical Prompt: We consider that an improved representation of an event prototype should encompass essential information from all the aforementioned prompts. Therefore we propose *United Lexical Prompt*, which is semantically rich by merging Event Type Name, Frame Names, Lexical Units and Seed Triggers. For example, The United prompt for Attack event type is *attack counterattack invading suicide_attack savaged penetrated sorties offensive attacker assault bombing*.

In our experiments, we automatically extract event type names from MAVEN [20], a designated event ontology. Prototype seed triggers for each event type are also automatically selected from the annotated data. The frame names, frame definition, Lexical Units and United Lexical Prompt are all obtained by aligning event type name to FrameNet via heuristic rules.

IV. METHODOLOGY

We now present our prompt-enhanced prototype framework for Few-shot Event Detection. The overall structure of our method is shown in Figure 2. Next, we will describe the details of this framework.

Constructing Prompts Given a linguistic corpus FrameNet $R = \{r_1, r_2, ..., r_N\}$, we take each event type t as a query to match the corresponding frames. Specifically, we first use a pre-trained SBERT [32] encoder to encode the whole frames from FrameNet and event type t to get their representations $\mathbf{R} = \{\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_N\}$ as well as $\mathbf{t}$. We then calculate the cosine similarity of $\mathbf{r}_i$ and $\mathbf{t}$:

$$S_c(\mathbf{r}_i, \mathbf{t}) = \frac{\mathbf{r}_i \cdot \mathbf{t}}{\|\mathbf{r}_i\| \|\mathbf{t}\|}$$

We select the top-K^4 ranked frames whose cosine similarity is larger than the threshold. The threshold is the average of the cosine similarity between the official annotation frames in MAVEN and the event type. Consequently, We automatically extract frame names from top-K^4 ranked frames and other frame-related prompts from top-K^1 ranked frames to obtain event prototype prompt $P = \{p_1, p_2, ..., p_N\}$.

Protoypical Representation Prototype source denotes the data or information source used in constructing the prototypes. The Aggregation form outlines how to aggregate or combine information from different prototype sources effectively to compute the proximity scores, i.e., $logits(y|x)$. In our experiments, We employ both prompts and event mentions as prototype sources, aggregating these two sources at loss-level [14]. This structure comprises multiple parallel branches b for each mention x, and each branch has its unique $logits^{(b)}(y|x)$, optimized with distinct loss components during training. Specifically, we allocate two branches for prompts and event mentions respectively.

For prompts branch, Given a prototype prompt $P = \{p_1^t, p_2^t, ..., p_N^t\}$ of event type t and event mention $x_i \in X$. We denote RoBERTa-base's [33] output representation of event mention x_i and prompt P as h_{x_i} and h_P respectively, where $h \in R^m$ and m represents the dimension of RoBERTa-base's hidden space.

For event mentions branch, we adopt MoCo Contrastive Learning [34] to aggregate event mentions. Given an event mention $x_i \in X$, x_i can serve as a prototype, and a given event type t may encompass multiple prototypes. Subsequently, we denote the representation of x_i as query and $\{x_j \mid (x_j \in X \wedge x_j \neq x_i \wedge y_j = y_i)\}$ as keys. In Moco CL, the query is encoded using the PLM's encoder, while the keys are encoded with a momentum encoder that is momentum-updated along the encoder for the query. Furthermore, it maintains a queue Q to store keys, reusing them when previously computed. We then get h_{x_i} and H_{x_j} which represent the query and key.

Event Detection With the aforementioned representations, At first, representations are transformed into the distance space using a transfer function to measure prototype proximity, i.e., normalization $f(h) = h/\|h\|_2$. Following that, we compute the distance between prototype and event mention using a distance function to obtain $logits(y|x)$, where the distance function adopts scaled cosine similarity $d_\mu(m, n) = -\frac{m^T n}{\mu}$:

$$logits^{(p)}(y|x) = -d_\mu(f(h_{x_i}), f(h_P))$$

$$logits^{(m)}(y|x) = \sum_{x_j \in Q} -\frac{d_T(f(h_{x_i}), f(H_{x_j}))}{|Q|}$$

$logits^{(p)}$ and $logits^{(m)}$ denote the proximity scores of prompts branch and event mentions branch respectively. Subsequently, we compute the losses for both branches and consolidate them for joint optimization:

$$P_{k \in \{p,m\}}^{(k)}(y|x) = Softmax_y[logits^{(k)}(y|x)]$$

$$L_{k \in \{p,m\}}^{(k)}(y|x) = -\sum_{(x,y)} y \log(P^{(k)}(y|x))$$

$$L = L^{(p)} + L^{(m)}$$

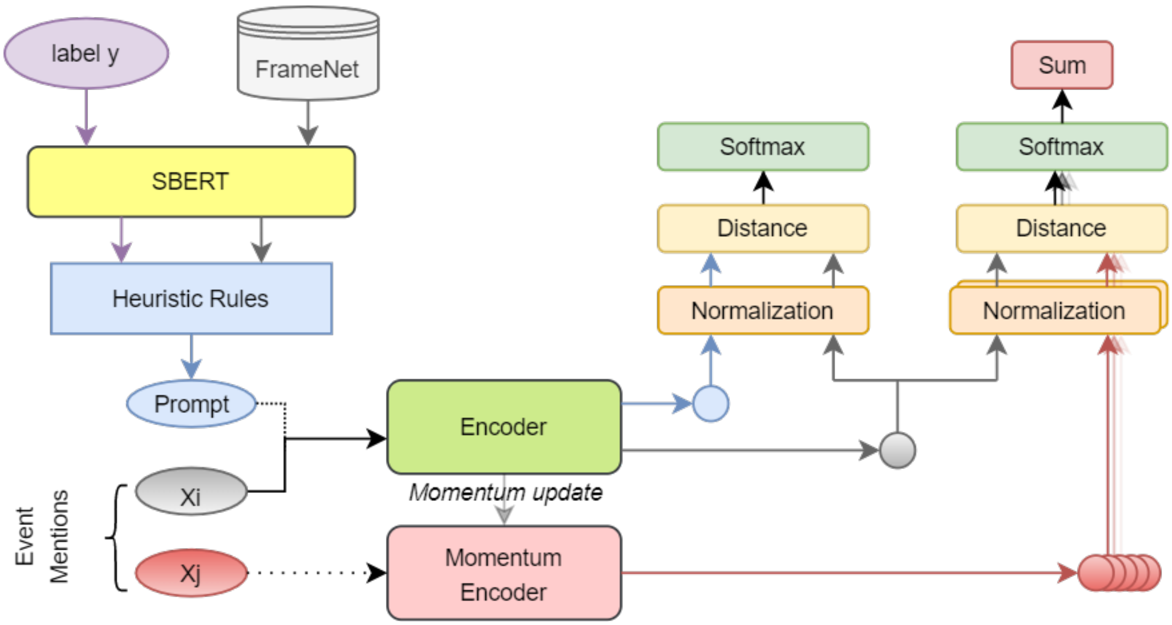

Fig. 2. Overall architecture of our model. For a given event mention x and event type y, this figure depicts the computation of the $logits(y|x)$. The symbol x_i represents the predicted event, and x_j denotes the set of event mentions sharing the event type y but differing from x_i. The blue circle, grey circle and red circle are representations of prompt, x_i and x_j respectively.

During inference, we implement nearest neighbor classification by assigning the sample to the event type with the closest proximity in the label sequence $Y = (T \cup N)$:

$$\hat{y}_x = \arg\min_{y \in Y} \min d(f(h_{x_i}), f(h_{proto}))$$

where h_{proto} is the representation of event prototype.

V. Experiment Setup

A. Dataset construction

We evaluate our method on a newly built large-scale event detection dataset MAVEN [20], which contains 4480 documents, 118732 event mention instances and 168 event types. Specific statistics for MAVEN are shown in Table I. We use the official split to construct original training and testing dataset for MAVEN dataset respectively.

TABLE I
Statistics of MAVEN

Maven Dataset	Train	Test
Event Type	168	
Sentences	32360	8035
Event Mentions	77993	18904

For few-shot settings, We sample sentences from original training dataset to construct training dataset and evaluation dataset and retain the original test set. For sampling, we adopt K-shot sampling strategy and employ the values K=(2,5,10) for the training set as well as K=(1,2,2) for the validation set. Following [14], [35], [36], we utilize a greedy sampling algorithm to approximately select K samples for each event type. Consequently, the actual number of samples for each event type may surpass K under this sampling strategy.

B. Implementation Details

For all experiments, we use the same pre-trained RoBERTa-base model and optimizing our model with AdamW [37]. The hyperparameters are as follows: warmup step 0.1, weight-decay coefficient 1e-5, maximum gradient norms 1.0, batch size 128, training steps 200, learning rate $\in$ [1e-5, 2e-5, 5e-5, 1e-4], scaled coefficient $\mu \in$ [0.1, 0.2, 0.3]. For MoCo setting, we keep a queue Q storing keys' representations with length 8192. In addition, all experiments are run on NVIDIA-V100 GPU.

C. Baselines and Evaluation

To examine the performance of our framework, we adopt state-of-the-art baseline methods for comparison. For prompt-based methods, we adopt EETE [25], PTE [7], UIE [26] and DEGREE [18]. For prototype-based methods, we choose ProtoNet [3], L-TapNet-CDT [17], CONTAINER [12] and FSED [14]. We briefly describe these methods as follows.

- **EETE**, a prompt-based method which converts few-shot ED task to natural language inference task.
- **PTE**, a cloze-style prompt method.
- **UIE**, a conditional generation-based prompt method.
- **DEGREE**, a variant of UIE by designing type-specific template.
- **ProtoNet**, a classical prototype-based method initially designed for episode learning.
- **L-TapNet-CDT**, a variant of ProtoNet by introducing TapNet and a collapsed dependency transfer (CDT) module.

- **CONTAINER**, a contrastive learning method for few-shot ED by treating mentions with the same event type as multiple prototypes.
- **FSED**, combinations of ProtoNet and contrastive learning.

For evaluation, we use the same metric as previous works [1], [14], [38], [39], i.e., micro-F1 score. Each experimental result value represents the average score and sample standard deviation of 10 sampled datasets for improving stability of results.

VI. Results and Discussion

Overall comparison Table II presents the overall performance comparison between different eight existing few-shot ED methods and our proposed framework. The optimal results are highlighted in bold. All results are averaged over ten repeated experiments, and sample standard deviations are presented in parentheses. We observe:(1) Our proposed framework achieves the best performance among all methods on MAVEN corpus under all three few-shot settings. Comparing with the previous state of the art, our method show up to 1.1% F1-score gain for 2-shot setting, 0.9% F1-score gain for 5-shot setting, and 1.1% F1-score gain for 10-shot setting. (2) Prompt-based methods have poorer performance compared to prototype-based methods, especially when given very few samples. It proves that defining effective prompts to conduct ED with limited annotations remains challenging. (3) Prototype-based methods shows encouraging performance with the introduction of relevant knowledge or methods to enhance the prototype representation. It also demonstrates the effectiveness of our approach to improving prototype representations using semantically rich prompts as well as contrastive learning.

TABLE II
Overall performance of eight existing few-shot ED methods and our proposed framework

Methods	2-shot	5-shot	10-shot
EETE	$15.7_{(0.6)}$	$19.1_{(0.3)}$	$21.4_{(0.2)}$
PTE	$38.4_{(4.2)}$	$42.6_{(7.2)}$	$49.8_{(1.9)}$
UIE	$29.3_{(2.9)}$	$38.3_{(4.2)}$	$43.4_{(3.5)}$
DEGREE	$40.0_{(2.9)}$	$45.5_{(3.2)}$	$48.5_{(2.1)}$
ProtoNet	$38.3_{(5.0)}$	$47.2_{(3.9)}$	$52.3_{(2.4)}$
L-TapNet-CDT	$43.2_{(3.8)}$	$49.8_{(2.9)}$	$53.5_{(3.4)}$
CONTAINER	$40.1_{(3.8)}$	$47.7_{(3.3)}$	$50.1_{(1.8)}$
FSED	$49.1_{(0.9)}$	$54.5_{(0.6)}$	$56.9_{(0.3)}$
Our framework	$\mathbf{50.2_{(1.1)}}$	$\mathbf{55.4_{(0.7)}}$	$\mathbf{58.0_{(0.3)}}$

Different prompts The overall performance of different prompts on MAVEN dataset is shown in Table III. The best results are marked bold, and the second best results are underlined. None means that no prompts branch is used to improve prototype representation. We can see:(1) The performance with or without prompts illustrates the necessity of prompts to enhance the prototype representation especially when the labelled samples are very scarce. (2) Lexical Prompts such as Event Type Name, Frame Names, Lexical Units and Seed Triggers have a similar performance in representing prototypes due to the fact that each specific prompt captures a distinct aspect of the event type semantics. So our proposed United Lexical Prompt integrates the semantic information of the above mentioned lexical prompts to describe event prototypes in a more comprehensive way. The superior performance of United Lexical Prompt can also demonstrate it. (3) Despite the encouraging achievements of the prototype-based approach using Frame Definition and Apex Prompt, they have a poorer performance compared to other prompts. We speculate that this is due to the fact that they contain redundant information unrelated to the event type.

TABLE III
Overall performance of different prompts

Prompts	2-shot	5-shot	10-shot
None	$40.5_{(1.4)}$	$47.7_{(0.9)}$	$50.6_{(0.5)}$
Event Type Name	$\underline{49.1_{(0.9)}}$	$\underline{54.5_{(0.6)}}$	$56.9_{(0.3)}$
Frame Names	$48.7_{(1.6)}$	$54.3_{(0.5)}$	$\underline{57.0_{(0.4)}}$
Frame Definition	$47.7_{(1.3)}$	$53.6_{(0.6)}$	$56.6_{(0.5)}$
Lexical Units	$\underline{49.1_{(1.1)}}$	$54.2_{(0.6)}$	$56.9_{(0.3)}$
Seed Triggers	$\underline{49.1_{(1.3)}}$	$\underline{54.5_{(0.7)}}$	$\underline{57.0_{(0.3)}}$
Apex Prompt	$48.3_{(1.3)}$	$53.9_{(0.6)}$	$56.8_{(0.3)}$
United Lexical Prompt	$\mathbf{50.2_{(1.1)}}$	$\mathbf{55.4_{(0.7)}}$	$\mathbf{58.0_{(0.3)}}$

Considering that lexical prompts are all composed of words or phrases, we further investigate the effect of constructing lexical prompts with different separators on the representation of prototypes. As shown in table IV, we choose comma, space, and or as separators for the experiments under three settings respectively. We found that the different separators used to represent the prototypes performed similarly, but using spaces as separators gave the best performance on average across the three settings.

TABLE IV
Overall performance of different separators

Separators	2-shot	5-shot	10-shot
or	50	55.2	57.9
comma	49.6	55.2	57.8
space	**50.2**	**55.4**	**58**

Case Study In this section, we select three representative prompts to present a case study under 2-shot setting. The prompts are Event Type Name, Frame Definition, and United Lexical Prompt which indicate individual lexical prompt, lexical prompt combined with related knowledge, and compound lexical prompt, respectively.

As shown in Table V, only United Lexical Prompt correctly predicts on all samples. We find that Frame Definition also predicts the correct result for the trigger word *impact*, suggesting that individual word combined with related knowledge (definition) can enrich semantic information in some way. However, for the trigger word *victory*, Event Type Name does not capture the event, and Frame Definition gives the wrong prediction. This shows that the redundant information carried in the definition may lead to a poor representation of the event prototype. For the trigger word *halting*, our prompt predicts the event *Process_end* event while the other two prompts predict *Hindering*. To a certain extent, both *halting* and *hindering* involve stopping. However, *halting* emphasises the sudden cessation of movement, behaviour or process, while *hindering* emphasises the negative impact of something on another, making it difficult to carry out or achieve a goal. *Process_end* is closer to the meaning of *halting*. This indicates that United Lexical Prompt provides a more comprehensive representation of the event prototype, whereas the other two prompts only represent certain aspects of the event.

TABLE V
CASE STUDY

Sentence	The victory of Pärnu and Salis had a direct impact on the further course of the war and contributed to halting the Swedish assault on Riga	
Trigger word	victory	
Gt type	Earnings_and_losses	
Prompts	Event Type Name	None
	Frame Definition	Getting
	United Lexical Prompt	Earnings_and_losses
Trigger word	impact	
Gt type	Influence	
Prompts	Event Type Name	None
	Frame Definition	Influence
	United Lexical Prompt	Influence
Trigger word	halting	
Gt type	Process_end	
Prompts	Event Type Name	Hindering
	Frame Definition	Hindering
	United Lexical Prompt	Process_end

VII. CONCLUSION

In this paper, We proposed an prompt-enhanced prototype framework to improve the representations of prototypes using prompts with rich semantics and contrastive learning. We further investigated various forms of prompts to represent event prototype and prove that describing event prototypes with comprehensive lexical prompts (United Lexical Prompt) improves the performance of few-shot ED. Extensive experiment results demonstrate the effectiveness of our approach by consistently outperforming state-of-the-art methods on MAVEN dataset. In the future, we aim to extend the model to the few-shot event argument extraction task and develop a joint model for few-shot event extraction task to solve the problem of insufficient annotation data in practical applications.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China under Grant (62372060), the NSFC-General Technology Basic Research Joint Funds under Grant (U1936220).

REFERENCES

[1] Y. Lin, H. Ji, F. Huang, and L. Wu, “A joint neural model for information extraction with global features,” in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 7999–8009.

[2] Z. Wang, X. Wang, X. Han, Y. Lin, L. Hou, Z. Liu, P. Li, J. Li, and J. Zhou, “CLEVE: Contrastive Pre-training for Event Extraction,” in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 6283–6297.

[3] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, p. 4080–4090.

[4] V. D. Lai, F. Dernoncourt, and T. H. Nguyen, “Exploiting the matching information in the support set for few shot event classification,” in *Advances in Knowledge Discovery and Data Mining: 24th Pacific-Asia Conference, PAKDD 2020, Singapore, May 11–14, 2020, Proceedings, Part II 24*. Springer, 2020, pp. 233–245.

[5] S. Deng, N. Zhang, J. Kang, Y. Zhang, W. Zhang, and H. Chen, “Meta-learning with dynamic-memory-based prototypical network for few-shot event detection,” in *Proceedings of the 13th International Conference on Web Search and Data Mining*, 2020, pp. 151–159.

[6] X. Cong, S. Cui, B. Yu, T. Liu, W. Yubin, and B. Wang, “Few-Shot Event Detection with Prototypical Amortized Conditional Random Field,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 28–40.

[7] T. Schick and H. Schütze, “It’s not just size that matters: Small language models are also few-shot learners,” in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 2339–2352.

[8] L. Huang, H. Ji, K. Cho, I. Dagan, S. Riedel, and C. Voss, “Zero-shot transfer learning for event extraction,” in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018, pp. 2160–2170.

[9] S. Shen, T. Wu, G. Qi, Y.-F. Li, G. Haffari, and S. Bi, “Adaptive knowledge-enhanced Bayesian meta-learning for few-shot event detection,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 2021, pp. 2417–2429.

[10] K. Zhao, X. Jin, L. Bai, J. Guo, and X. Cheng, “Knowledge-enhanced self-supervised prototypical network for few-shot event detection,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 6266–6275.

[11] S. Deng, N. Zhang, L. Li, C. Hui, T. Huaixiao, M. Chen, F. Huang, and H. Chen, “OntoED: Low-resource event detection with ontology embedding,” 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)*, 2021, pp. 2828–2839.

[12] S. S. S. Das, A. Katiyar, R. Passonneau, and R. Zhang, “CONTaiNER: Few-shot named entity recognition via contrastive learning,” in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 6338–6353.

[13] R. Zhang, W. Wei, X.-L. Mao, R. Fang, and D. Chen, “HCL-TAT: A hybrid contrastive learning method for few-shot event detection with task-adaptive threshold,” in *Findings of the Association for Computational Linguistics: EMNLP 2022*, 2022, pp. 1808–1819.

[14] Y. Ma, Z. Wang, Y. Cao, and A. Sun, “Few-shot event detection: An empirical study and a unified view,” in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023, pp. 11 211–11 236.

[15] X. Du and C. Cardie, “Event extraction by answering (almost) natural questions,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 671–683.

[16] R. Feng, J. Yuan, and C. Zhang, “Probing and fine-tuning reading comprehension models for few-shot event extraction,” *arXiv preprint arXiv:2010.11325*, 2020.

[17] Y. Hou, W. Che, Y. Lai, Z. Zhou, Y. Liu, H. Liu, and T. Liu, "Few-shot slot tagging with collapsed dependency transfer and label-enhanced task-adaptive projection network," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 1381–1393.

[18] I.-H. Hsu, K.-H. Huang, E. Boschee, S. Miller, P. Natarajan, K.-W. Chang, and N. Peng, "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*, 2022, pp. 1890–1908.

[19] S. Li, L. Liu, Y. Xie, H. Ji, and J. Han, "P4e: Few-shot event detection as prompt-guided identification and localization," *arXiv preprint arXiv:2202.07615*, 2022.

[20] X. Wang, Z. Wang, X. Han, W. Jiang, R. Han, Z. Liu, J. Li, P. Li, Y. Lin, and J. Zhou, "Maven: A massive general domain event detection dataset," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, 2020, pp. 1652–1671.

[21] H. Ji and R. Grishman, "Refining event extraction through cross-document inference," in *Proceedings of ACL-08: Hlt*, 2008, pp. 254–262.

[22] Q. Li, H. Ji, and L. Huang, "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)*, 2013, pp. 73–82.

[23] K. Cao, X. Li, M. Fan, and R. Grishman, "Improving event detection with active learning," in *Proceedings of the International Conference Recent Advances in Natural Language Processing*, 2015, pp. 72–77.

[24] T. Zhang, S. Whitehead, H. Zhang, H. Li, J. Ellis, L. Huang, W. Liu, H. Ji, and S.-F. Chang, "Improving event extraction via multimodal integration," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 270–278.

[25] Q. Lyu, H. Zhang, E. Sulem, and D. Roth, "Zero-shot event extraction via transfer learning: Challenges and insights," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2021, pp. 322–332.

[26] Y. Lu, Q. Liu, D. Dai, X. Xiao, H. Lin, X. Han, L. Sun, and H. Wu, "Unified structure generation for universal information extraction," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 5755–5772.

[27] T. H. Nguyen and R. Grishman, "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)*, 2015, pp. 365–371.

[28] M. V. Nguyen, V. D. Lai, and T. H. Nguyen, "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*, 2021, pp. 27–38.

[29] S. Wang, M. Yu, and L. Huang, "The art of prompting: Event detection based on type specific prompts," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 2023, pp. 1286–1299.

[30] C. F. Baker, C. J. Fillmore, and J. B. Lowe, "The berkeley framenet project," in *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*, 1998, pp. 86–90.

[31] S. Wang, M. Yu, S. Chang, L. Sun, and L. Huang, "Query and extract: Refining event extraction as type-oriented binary decoding," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 169–182.

[32] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019, pp. 3982–3992.

[33] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[34] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 9729–9738.

[35] Y. Yang and A. Katiyar, "Simple and effective few-shot named entity recognition with structured nearest neighbor learning," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6365–6375.

[36] J. Ma, M. Ballesteros, S. Doss, R. Anubhai, S. Mallya, Y. Al-Onaizan, and D. Roth, "Label semantics for few shot named entity recognition," in *Findings of the Association for Computational Linguistics: ACL 2022*, 2022, pp. 1956–1971.

[37] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *International Conference on Learning Representations*, 2017. [Online]. Available: https://api.semanticscholar.org/CorpusID:53592270

[38] Y. Chen, L. Xu, K. Liu, D. Zeng, and J. Zhao, "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)*, 2015, pp. 167–176.

[39] T. H. Nguyen, K. Cho, and R. Grishman, "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*, 2016, pp. 300–309.