Class-Incremental Few-Shot Event Detection

Kailin Zhao[1,2], **Xiaolong Jin**[1,2*], **Long Bai**[1], **Jiafeng Guo**[1,2], **Xueqi Cheng**[1,2]
[1]Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences;
[2]School of Computer Science and Technology, University of Chinese Academy of Sciences
{zhaokailin17z, jinxiaolong, bailong18b, guojiafeng, cxq}@ict.ac.cn

Abstract

Event detection is one of the fundamental tasks in information extraction and knowledge graph. However, a realistic event detection system often needs to deal with new event classes constantly. These new classes usually have only a few labeled instances as it is time-consuming and labor-intensive to annotate a large number of unlabeled instances. Therefore, this paper proposes a new task, called class-incremental few-shot event detection. Nevertheless, this task faces two problems, i.e., old knowledge forgetting and new class overfitting. To solve these problems, this paper further presents a novel knowledge distillation and prompt learning based method, called Prompt-KD. Specifically, to handle the forgetting problem about old knowledge, Prompt-KD develops an attention based multi-teacher knowledge distillation framework, where the ancestor teacher model pre-trained on base classes is reused in all learning sessions, and the father teacher model derives the current student model via adaptation. On the other hand, in order to cope with the few-shot learning scenario and alleviate the corresponding new class overfitting problem, Prompt-KD is also equipped with a prompt learning mechanism. Extensive experiments on two benchmark datasets, i.e., FewEvent and MAVEN, demonstrate the superior performance of Prompt-KD.

Keywords: Information Extraction, Knowledge Discovery/Representation, Text Mining

1. Introduction

Event detection is one of the fundamental tasks in information extraction and knowledge graph, which specifically extracts trigger words from texts indicating the occurrence of events and further classifies them into different event classes. For example, in "*Tom was injured by falling rocks*", the trigger word is *"injured"*, indicating an *Injure* event. Event detection benefits many downstream applications, e.g., event graph construction, question answering and information retrieval.

A realistic event detection system often needs to deal with new classes of events, which continuously arrive. Nevertheless, these new classes usually have only a few labeled instances as it is time-consuming and labor-intensive to annotate a large number of unlabeled instances. Therefore, how to incrementally learn the new event classes with only a few labeled instances has become a challenging problem to the event detection system. To address this problem, in this paper we propose the Class-Incremental Few-Shot Event Detection (CIFSED) task.

Existing Few-Shot Event Detection (FSED) methods have achieved satisfying performance (Cong et al., 2021; Zhao et al., 2022). Therefore, a straightforward method for CIFSED is to train these FSED methods on base classes and fine-tune them on new classes. However, if directly applying these methods in the CIFSED scenario via simple fine-tuning, two severe problems will emerge (Tao et al., 2020b): 1) old knowledge forgetting: The model will forget old knowledge when dealing with new event classes and thus lower its own performance; 2) new class overfitting: The model is prone to overfitting to new classes and thus shows poor generalization ability on subsequent classes, which is caused by the few-shot scenarios. Therefore, the primary objective of a CIFSED method is to learn new classes while maintaining old knowledge.

In order to achieve the similar objective in other fields such as image classification and named entity recognition, two kinds of Class-Incremental Few-Shot Learning (CIFSL) methods have been proposed, i.e., topological structure based methods (Tao et al., 2020a,b) and knowledge distillation based methods (Cheraghian et al., 2021; Dong et al., 2021; Wang et al., 2022). The former kind of methods preserve the old knowledge by maintaining the topology of the feature space in the network. The latter kind of methods maintain the output probabilities corresponding to the learned classes by adapting the model obtained from the last step based on new classes. Although this kind of methods have become the mainstream ones recently, they are defective in overcoming the old knowledge forgetting problem as this step-by-step manner causes the model to deviate from base knowledge as time goes by (Dong et al., 2021). Furthermore, the new class overfitting problem has not been paid much attention (Tao et al., 2020b,a).

To solve the above challenging problems, we propose a novel Knowledge Distillation and Prompt learning based method, called Prompt-KD, for CIFSED. Prompt-KD presents an attention based multi-teacher knowledge distillation framework.

*Corresponding author.

LREC-COLING 2024, pages 3261–3270

Therein, the ancestor teacher model, trained on base classes, is employed to derive the student model in the first learning session. From the second learning session onwards, the father teacher model, which is actually the student model in the last learning session, derives the new student model via adaptation. This framework also adopts an attention mechanism to balance the different importance between these two teacher models as to the student model. To ease the forgetting problem about base knowledge, the ancestor teacher model is reused constantly in all learning sessions. Furthermore, Prompt-KD employs a prompt learning mechanism with additional predefined texts (i.e., prompts) to the input instances in the support set, so as to cope with the few-shot learning scenario and alleviate the corresponding new class overfitting problem.

In summary, the main contributions of this paper are three-fold.

- We propose for the first time, to the best of our knowledge, the Class-Incremental Few-Shot Event Detection (CIFSED) task, which often exists in the real-world event detection systems.
- We propose a novel knowledge distillation and prompt learning based method, called Prompt-KD, for CIFSED. To handle the forgetting problem about old knowledge, Prompt-KD presents an attention based multi-teacher knowledge distillation framework. On the other hand, in order to cope with the few-shot learning scenario and alleviate the corresponding new class overfitting problem, Prompt-KD is also equipped with a prompt learning mechanism.
- Extensive experiments on two benchmark datasets, i.e., FewEvent and MAVEN, demonstrate the superior performance of Prompt-KD.

2. Related Works

2.1. Class-incremental Few-shot Learning

As aforesaid, there are two kind of approaches to the CIFSL task, i.e., topological structure based and knowledge distillation based, respectively. Topological structure based CIFSL methods preserve the old knowledge by maintaining the topology of the feature space in the network. Tao et al. (2020b) were the first to propose the CIFSL task, and further presented a framework, called TOPIC, which adopts a neural gas network to learn feature space topology for knowledge representation. Next, Tao et al. (2020a) proposed a new TPCIL framework, which employs an elastic Hebbian graph to model the feature space topology. Zhang et al. (2021) adopted a decoupled training strategy for representation learning and classifier learning to ease the old knowledge forgetting problem. Knowledge distillation based CIFSL methods maintains the output probabilities corresponding to the learned classes. Cheraghian et al. (2021) introduced semantic information into knowledge distillation and proposed a semantically-guided framework. Later, Dong et al. (2021) put forward a relation knowledge distillation framework, which constrains the relations among instances rather than their absolute positions. To address the CIFSED task, we propose a novel attention based multi-teacher knowledge distillation framework, which can repeatedly employ the ancestor teacher model to handle the forgetting probl abouemt base knowledge.

2.2. Event Detection

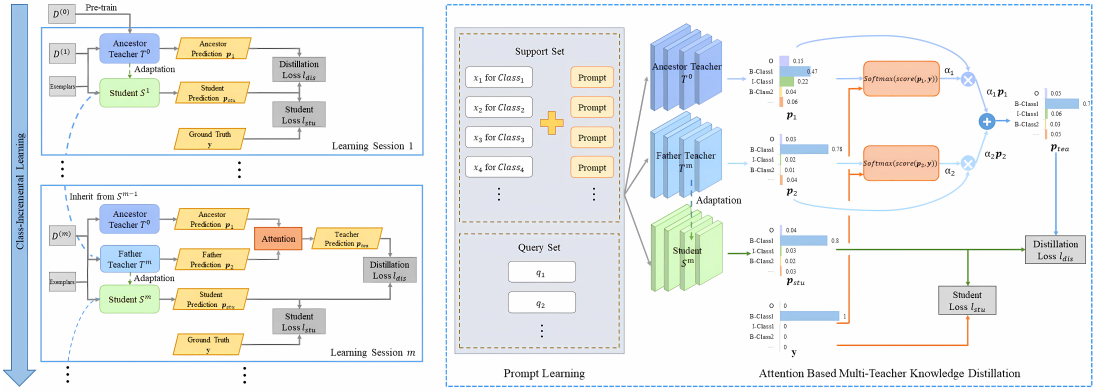
There are two kinds of approaches to event detection, i.e., pipeline ones and joint ones. Pipeline approaches follow the identification-then-classification process and thus suffer from the error propagation problem. Due to this reason, joint approaches have attracted much attention. Under the few-shot scenarios, Cong et al. (2021) proposed PA-CRF based on a sequence tagging method. Under the class-incremental scenarios, Cao et al. (2020), Yu et al. (2021) and Liu et al. (2022b) solved event detection based on the knowledge distillation framework. In this paper, we choose the first and representative joint FSED model, i.e., PA-CRF, as our base model.

2.3. Prompt Learning

Prompt learning aims to minimize the gap between the pre-training objective and the downstream fine-tuning objective. Brown et al. (2020) proposed GPT-3, which is the first to employ prompts for downstream tasks without introducing extra parameters, breaking the traditional pre-training and fine-tuning mode. Later, prompt learning has been applied in many information extraction tasks, e.g., entity extraction (Cui et al., 2021; Liu et al., 2022a; Ding et al., 2021) and relation extraction (Han et al., 2022). Recently, Li et al. (2022) introduced prompt learning into FSED and designed the cloze prompt as well as class-aware prompt for event class identification and trigger localization, respectively. In this paper, we design a new cloze prompt for joint FSED methods, which can tackle the error propagation challenge.

3. Problem Formulation

Inspired by existing CIFSL works in other fields (Dong et al., 2021; Tao et al., 2020b; Wang et al., 2022), we formulate CIFSED as follows. In



(a) The class-incremental learning process diagram of Prompt-KD. (b) The detailed diagram of Prompt-KD in the m-th learning session ($1<m<M$).

Figure 1: The diagram of the Prompt-KD method.

CIFSED, we assume that a series of datasets $D^{(0)}$, $D^{(1)}$ ... $D^{(M)}$ constantly arrive, each of which is to be handled in a learning session. Here, $D^{(0)}$ is the large-scale dataset with base classes and $D^{(m)}(1 \leq m \leq M)$ is the m-th few-shot dataset containing new classes. All $D^{(m)}(0 \leq m \leq M)$ have disjoint event class set with each other. Following the episodic learning strategy (Laenen and Bertinetto, 2021), there are several episodes in the 0-th learning session and only one episode in the m-th $(1 \leq m \leq M)$ learning session. For each episode, a support set and a query set are randomly instanced from $D^{(m)}(0 \leq m \leq M)$, which is formulated in the N-way K-shot paradigm. Given the support set $S = \{(x_i, y_i)\}_{i=1}^{N \times K}$ which has N classes and each class has K labeled instances, FSED aims to predict the labels of tokens in the query set Q. In the support set S, $x_i = \{w_i^1, w_i^2, ..., w_i^n\}$ denotes an n-word sequence, and $y_i = \{l_i^1, l_i^2, ..., l_i^n\}$ denotes its corresponding label sequence of tokens. In the query set $Q = \{q_i\}_{i=1}^{N \times U}$, each class contains U unlabeled instances, where q_i refers to a sequence of unlabeled tokens. Since joint FSED is formulated as a sequence tagging process, the label l_i consists of two parts: the position part and the type part. For the position part, there are three types, i.e., B, I and O. B and I indicate that the corresponding word is the beginning and inside word of the event trigger, respectively, which may contain multiple words. O indicates that corresponding word does not belong to any trigger. Therefore, the total number of token labels is $2N + 1$ (N for *B-Class*, another N for *I-Class*, and 1 for label O).

4. The Prompt-KD Method

An illustrative diagram of the Prompt-KD method is presented in Figure 1, where the left part (a) illustrates the class-incremental learning process of Prompt-KD, while the right part (b) presents a more detailed implementation of the m-th learning session. As we can see, Prompt-KD consists of two main modules in each learning session, i.e., the attention based multi-teacher knowledge distillation module and the prompt learning module. The former module takes the prompt concatenated support instances and query instances as its inputs, and produces the prediction probabilities of the query tokens. The latter module aims to add instructions to the support instances to help the training process and outputs the enhanced instances with prompt.

4.1. Attention Based Multi-Teacher Knowledge Distillation

To address the old knowledge forgetting problem, this module presents a two-teacher one-student knowledge distillation framework and further employs an attention mechanism so as to balance the different importance between these two teacher models. This module contains five main components, i.e., the exemplars, the two teacher models, the attention mechanism, the student model and the loss functions. The exemplars are selected in each learning session to replay the instances of old classes. They are taken by Prompt-KD as its input, together with the instances from new classes. The two teacher models produce their prediction probabilities respectively by inputting the prompt enhanced support instances and query instances. The attention mechanism takes the above probabilities as its input and calculates the weighted teacher probabilities. The student model takes the same input as the teacher models and outputs the student probabilities. The loss functions, including the distillation loss and the student loss, are employed to update the parameters of Prompt-KD.

4.1.1. The Exemplars

Generally, each learning session has its own exemplars, which are selected from the learned classes (Dong et al., 2021). For example, in the

m-th learning session ($m > 0$), the exemplars are obtained from the instances in $D^{(0)}$, ..., $D^{(m-1)}$. Specifically, from $D^{(0)}$, a few randomly selected instances are adopted as the exemplars of base classes. Since all $D^{(i)}$ ($1 \leq i \leq m-1$) have few-shot instances (e.g., 1-shot or 3-shot), they are all taken as the exemplars of the corresponding classes. Then, in the m-th learning session, Prompt-KD takes $D^{(m)}$ together with the corresponding exemplars as its input, which contains a new support set S' and a new query set Q'.

4.1.2. The Two Teacher Models

To overcome the old knowledge forgetting problem, Prompt-KD adopts two teacher models, i.e., the ancestor teacher model T^0 and the father teacher model $T^m (m > 1)$. The former refers to the model pre-trained on the large-scale dataset $D^{(0)}$, whilst the latter is actually the student model $S^{m-1} (m > 1)$ in the last learning session, as shown in Figure 1(a). In this paper, we adopt the pre-trained PA-CRF as the ancestor teacher model for FSED, which mainly consists of four units, i.e., encoder unit, emission unit, transition unit and decoder unit.

The encoder unit takes the instances in the support set S' and the query set Q' as its input and maps them into the embedding space to represent their semantic meanings. Given an input $x_i = \{w_i^1, w_i^2, ..., w_i^{n+P}\}$, BERT-base-uncased (Kenton and Toutanova, 2019) is employed to get its embeddings as $\boldsymbol{x}_i = \{\boldsymbol{w}_i^1, \boldsymbol{w}_i^2, ..., \boldsymbol{w}_i^{n+P}\} = BERT(x_i)$, where $\boldsymbol{w}_i^j$ denotes the representation of token w_i^j, which is of H dimension, and P is the length of the prompt. Thus, the support instance embedding set $\boldsymbol{S}'$ can be formulated as

$$\boldsymbol{S}' = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_{(m \times N + B) \times K}\}, \tag{1}$$

where B is the number of event classes in $D^{(0)}$.

Similarly, the query instance embedding set $\boldsymbol{Q}'$ is formulated as

$$\boldsymbol{Q}' = \{\boldsymbol{q}_1, \boldsymbol{q}_2, ..., \boldsymbol{q}_{(m \times N + B) \times U}\}, \tag{2}$$

where $\boldsymbol{q}_i$ denotes the embedding representation of q_i by $\boldsymbol{q}_i = BERT(q_i)$.

The emission unit takes the representations $\boldsymbol{S}'$ and $\boldsymbol{Q}'$ as its input and calculates the prototype $\boldsymbol{c}_l$ to each label l of tokens based on $\boldsymbol{S}'$ as

$$\boldsymbol{c}_l = \frac{1}{|W(\boldsymbol{S}', l)|} \sum_{\boldsymbol{w} \in W(\boldsymbol{S}', l)} \boldsymbol{w}, \tag{3}$$

where $W(\boldsymbol{S}', l)$ indicates the token set with label l in $\boldsymbol{S}'$ and $\boldsymbol{w}$ is the representation of a token in it. Then, this unit calculates the similarities between the presentations of the query tokens and the prototypes as emission scores. In practice, the dot product operation is chosen to measure the similarity.

The transition unit takes the representations of the prototypes as its input and then generates the parameters (i.e., mean and variance) of Gaussian distribution as the transition scores.

Based on the above obtained emission scores and transition scores, the decoder unit calculates the probabilities of possible label sequences for the given tokens in the query set Q' and then derives the predicted label sequences. The Monte Carlo sampling technique (Gordon et al., 2019) is employed to approximate the integral. In the inference phase, the first teacher model T^0 and the second one T^m adopt the Viterbi algorithm (Forney, 1973) to decode the probability distributions $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ to different label sequences for the query tokens, respectively. The event detection process for the two teacher models can be simplified as

$$\boldsymbol{p}_1 = T^0(S', Q'), \boldsymbol{p}_2 = T^m(S', Q'). \tag{4}$$

4.1.3. The Attention Mechanism

The attention mechanism is employed to balance the different importance between the two teacher models. The final probability distribution of the teacher models is denoted as $\boldsymbol{p}_{tea}$, which is calculated by

$$\boldsymbol{p}_{tea} = \sum_{i=1}^{2} \alpha_i \boldsymbol{p}_i, \tag{5}$$

where α_i refers to the weight of the i-th teacher model. Moreover, α_i is obtained via $\alpha_i = Softmax(\boldsymbol{s}_i)$. Therein, $\boldsymbol{s}_i$ is calculated as

$$\boldsymbol{s}_i = score(\boldsymbol{p}_i, \boldsymbol{y}) = \boldsymbol{p}_i \boldsymbol{W} \boldsymbol{y}, \tag{6}$$

where $\boldsymbol{y}$ denotes the ground truth distribution of the query tokens to different label sequences and $\boldsymbol{W}$ is a learnable matrix.

4.1.4. The Student Model

As shown in Figure 1(a), in the first learning session, the student model S^1 is derived from the ancestor teacher model T^0. In the subsequent learning sessions, S^m is obtained from the father teacher model T^m via adaptation based on the support set S'. The adaptation process is formulated as

$$S^m = T^m(S'). \tag{7}$$

The same as the event detection process of the teacher models, the probability distribution $\boldsymbol{p}_{stu}$ of the student model to different label sequences for the query tokens is calculated as

$$\boldsymbol{p}_{stu} = S^m(S', Q'). \tag{8}$$

Stage 1	This is a [mask] event.[SEP] Its trigger words are [mask].
Stage 2	This is a [mask] event, which is learned [mask*].[SEP] Its trigger words are [mask].
Stage 3	This is a [mask] event, which is learned [mask*].[SEP] Its trigger words are [mask].

Table 1: The three-stage curriculum learning based prompts.

4.1.5. The Loss Functions

Since Prompt-KD adopts the knowledge distillation framework, its loss functions consist of two parts, i.e., distillation loss and student loss. The distillation loss l_{dis} is obtained upon the cross entropy loss function $L(.,.)$ between the the probability distributions $\boldsymbol{p}_{tea}$ and $\boldsymbol{p}_{stu}$ as

$$l_{dis} = L(\boldsymbol{p}_{tea}, \boldsymbol{p}_{stu}). \tag{9}$$

In the meantime, the student loss l_{stu} is similarly calculated via $L(.,.)$ as

$$l_{stu} = L(\boldsymbol{p}_{stu}, \boldsymbol{y}). \tag{10}$$

Then, the final loss l is obtained via summing the above two losses:

$$l = l_{dis} + l_{stu}. \tag{11}$$

4.2. Prompt Learning

To cope with the few-shot learning scenario and alleviate the corresponding new class overfitting problem, Prompt-KD adopts prompt learning as it can concatenate instructions with semantic information after the support instances and thus reduces the dependence on labeled instances. Specifically, Prompt-KD presents a new cloze prompt for joint FSED, which is designed as "This is a [mask] event.[SEP] Its trigger words are [mask].".

Moreover, this module is equipped with a prompt-oriented curriculum learning mechanism. To the best of our knowledge, this paper is the first to combine curriculum learning with prompt learning. In detail, a three-stage curriculum learning based prompt is adopted, as shown in Table 1. At Stage 2, the candidate set of [mask*] is {"before", "now"}, where "before" indicates that the event class has been learned in the past, while "now" denotes the event class is being learned at present. At Stage 3, {"before", "recently", "now"} is employed as the candidate set, where "before" denotes that the event class has been learned a long time ago, "recently" suggests that the event class has been learned a while ago, and "now" is the same as that at Stage 2.

In practice, dealing with a CIFSED task with M learning sessions, Stages 1-3 contain the sessions 1 to $\lceil \frac{1}{3}M \rceil$, $\lceil \frac{1}{3}M + 1 \rceil$ to $\lceil \frac{2}{3}M \rceil$, and $\lceil \frac{2}{3}M + 1 \rceil$ to M, respectively, where $\lceil \cdot \rceil$ denotes rounding up to an integer.

5. Experiments

5.1. Datasets and Evaluation Metrics

We conduct experiments on two FSED benchmark datasets, i.e., FewEvent (Deng et al., 2020) and MAVEN (Wang et al., 2020), which contain 100 and 168 classes, respectively.

FewEvent. FewEvent is designed as a benchmark FSED dataset, which extracts event classes in ACE-2005 (Doddington et al., 2004) and TAC-KBP-2017 (Ji and Grishman, 2011). Besides, it also extends many new event classes from Wikipedia and Freebase via automatic tagging. The dataset contains a total of 70,852 instances, with 19 event classes subdivided into 100 subclasses, where each subclass has an average of about 700 instances.

MAVEN. MAVEN is a large-scale common domain event detection dataset that contains 4480 documents and 118732 event instances covering 168 event classes. For MAVEN, we adopt 100 classes that have more than 200 instances, following the previous work (Zhao et al., 2022).

We set up four configurations, namely, 5-way 1-shot, 5-way 3-shot, 10-way 1-shot and 10-way 3-shot, for each CIFSED task. We follow the evaluation protocols in (Tao et al., 2020b; Dong et al., 2021) and thus redivide the above two datasets. For both FewEvent and MAVEN, 50 classes are randomly selected as base classes for $D^{(0)}$ and the other 50 classes are equally split for class-incremental learning. For the 5-way tasks, the 50 classes especially for class-incremental learning are equally divided into 10 subsets. Therefore, we obtain 11 subsets (i.e., $D^{(0)}, D^{(1)}, ..., D^{(10)}$) totally, where each $D^{(m)}(m > 0)$ has 5 classes and each class contains randomly sampled 1 or 3 support instances and 1 query instances. Similar with the 5-way tasks, 6 subsets (i.e., $D^{(0)}, D^{(1)}, ..., D^{(5)}$) are obtained for the 10-way tasks and each $D^{(m)}(m > 0)$ contains 10 classes.

In addition, we adopt the standard micro F1 score as the evaluation metric and report the averages upon 5 randomly initialized runs.

5.2. Implementation Details and Parameter Setting

BERT-base-uncased (Kenton and Toutanova, 2019) is employed as the encoder for both the teacher models and the student model, whose in-

Dataset: FewEvent												
Method	Learning Sessions											Average
	0	1	2	3	4	5	6	7	8	9	10	F1
PA-CRF-Meta	78.19	60.67	58.80	56.25	55.82	54.72	53.82	52.67	51.82	46.62	44.78	55.83
PA-CRF-CIL	78.19	55.95	47.36	45.90	43.03	41.36	39.78	37.78	36.79	35.09	32.78	44.91
Prompt-KD	78.19	60.58	51.78	50.31	49.69	50.53	48.61	47.57	47.32	47.03	45.22	52.44
Dataset: MAVEN												
Method	Learning Sessions											Average
	0	1	2	3	4	5	6	7	8	9	10	F1
PA-CRF-Meta	73.04	42.83	38.97	37.92	35.66	33.33	32.72	29.83	29.51	26.05	25.46	36.84
PA-CRF-CIL	73.04	35.02	30.93	28.60	26.23	25.00	24.14	23.07	22.61	21.68	18.96	29.93
Prompt-KD	73.04	39.47	36.98	35.07	32.98	30.03	30.05	28.02	27.15	25.89	23.71	34.76

Table 2: The F1 scores (%) of the 5-way 1-shot tasks on two benchmark datasets: FewEvent and MAVEN.

Dataset: FewEvent							
Method	Learning Sessions						Average
	0	1	2	3	4	5	F1
PA-CRF-Meta	77.84	61.95	58.23	54.88	54.58	54.13	60.26
PA-CRF-CIL	77.84	50.17	44.49	33.04	28.88	23.31	42.95
Prompt-KD	77.84	55.48	47.69	40.79	34.09	32.05	47.99
Dataset: MAVEN							
Method	Learning Sessions						Average
	0	1	2	3	4	5	F1
PA-CRF-Meta	69.04	31.58	26.26	24.94	20.68	19.38	31.98
PA-CRF-CIL	69.04	26.85	21.17	16.67	14.47	11.69	26.64
Prompt-KD	69.04	28.40	24.07	19.78	18.23	16.95	30.88

Table 3: The F1 scores (%) of the 10-way 1-shot tasks on two datasets: FewEvent and MAVEN.

put sentence has max length of 128 and the hidden size H is 768. Prompt-KD is trained with the 1e-5 learning rate with the AdamW optimizer. Moreover, the dropout is 0.1 and the batch size is 1. We pretrain PA-CRF with 10,000 episodes on $D^{(0)}$ as the ancestor teacher model. Furthermore, we evaluate the performance on $D^{(0)} \cup D^{(1)} \cup ... \cup D^{(m)}$ in the m-th learning session, all following the episodic paradigm. We run all experiments using PyTorch 1.5.1 on the Nvidia V100 GPU with 32GB memory, Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz with 128GB memory on CentOS Linux release 7.9.2009 (Core).

5.3. Baseline Models

Since we are the first to propose the CIFSED task, there are no existing methods for it. In order to investigate the validity and effectiveness of Prompt-KD, we develop two variants (i.e., PA-CRF-CIL and PA-CRF-Meta) of the first and representative joint FSED method, i.e., PA-CRF, for experimental comparison, as it is the base model of Prompt-KD. Specifically, PA-CRF-CIL is obtained by applying the PA-CRF model in the class-incremental scenario, where the model in the m-th learning session is derived via fine-tuning the model from the last learning session based on $D^{(m)}$ and the corresponding exemplars. PA-CRF-Meta is trained via episodic learning under the meta learning framework on the joint dataset $D^{(0)} \cup D^{(1)} \cup ... \cup D^{(m)}$, which can thus be regarded, to a certain degree, as an upper bound model.

5.4. Experimental Results

Tables 2 and 3 present the overall experimental results on the 5-way 1-shot and the 10-way 1-shot tasks, whilst Figure 2 compares the test F1 scores of the 5-way 3-shot and the 10-way 3-shot tasks on FewEvent and MAVEN, respectively. We summarize the results as follows.

- Our Prompt-KD method outperforms the PA-CRF-CIL baseline and achieves the state-of-the-art performance consistently on both datasets, all tasks and all learning sessions. Particularly, the average F1 score of Prompt-KD increases by 5-8% on FewEvent and 4-5% on MAVEN, respectively, compared to PA-CRF-CIL.

- As shown in Figure 2, the F1 score curve of Prompt-KD is close to that of PA-CRF-Meta on MAVEN, where these curves seem almost overlapping. However, there exists a clear distance between the F1 score curves of Prompt-KD and PA-CRF-Meta on FewEvent, which

Method	KD	AT	ATT	PL	CL	Learning Sessions					Average
						1	2	3	4	5	F1
Prompt-KD	√	√	√	√	√	**55.48**	**47.69**	**40.79**	**34.09**	**32.05**	**42.02**
$\mathcal{A}$	√	√	√	√		55.48	47.34	40.33	33.79	31.51	41.69
$\mathcal{B}$	√	√	√			54.36	46.12	38.02	30.19	27.77	39.29
$\mathcal{C}$	√	√				51.64	45.98	37.49	30.01	26.93	38.41
$\mathcal{D}$	√					51.64	45.38	34.60	29.57	24.94	37.23
PA-CRF-CIL						50.17	44.49	33.04	28.88	23.31	35.97

Table 4: The results of the ablation study on the 10-way 1-shot tasks on FewEvent.

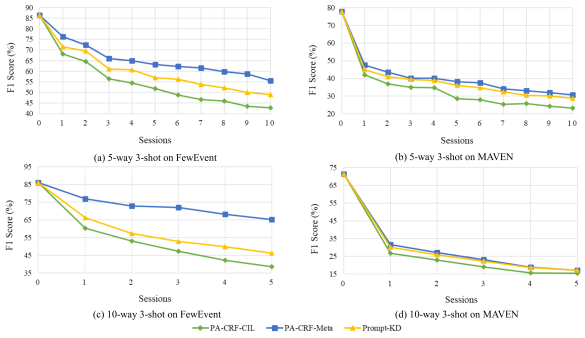


Figure 2: The F1 score curves of the 5-way 3-shot and the 10-way 3-shot tasks on FewEvent and MAVEN.

indicates that the CIFSED method has great potential to be improved on FewEvent.

- The average F1 scores of Prompt-KD and PA-CRF-CIL on FewEvent on the 5-way tasks are higher than those on the 10-way tasks, as shown in Tables 2 and 3 as well as Figure 2. This phenomenon indicates that learning with fewer learning sessions with more classes suffers from a more serious old knowledge forgetting problem than that with more learning sessions with fewer classes.

- As shown in Table 2, in Sessions 9 and 10 on the 5-way 1-shot tasks on FewEvent, our Prompt-KD method even outperforms PA-CRF-Meta, which can further demonstrate the ability of Prompt-KD to overcome the old knowledge forgetting problem. This phenomenon may be due to that PA-CRF-Meta is not a theoretical upper bound model, where although the meta learning framework can alleviate the class imbalance problem, it also damages the performance of the PA-CRF-Meta model on base classes with a large number of instances.

5.5. Ablation Study

We conduct ablation studies to investigate the effectiveness of Knowledge Distillation (KD), Ancestor Teacher (AT), Attention (ATT), Prompt Learning (PL) and Curriculum Learning (CL), as well as their impacts on the performance of Prompt-KD on the 10-way 1-shot tasks. Without loss of generality, these ablation studies are carried out on FewEvent. Specifically, the ablated models of Prompt-KD incrementally without CL, (PL and CL), (ATT, PL and CL), (AT, ATT, PL and CL) are identified as $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and $\mathcal{D}$, respectively. As shown in Table 4, the performance of the ablated model $\mathcal{A}$ falls compared to that of Prompt-KD, which indicates that CL contributes to the effectiveness of Prompt-KD. Similarly, the comparisons between the results of the ablated models $\mathcal{A}$ and $\mathcal{B}$, $\mathcal{B}$ and $\mathcal{C}$, $\mathcal{C}$ and $\mathcal{D}$, as well as $\mathcal{D}$ and PA-CRF-CIL, demonstrate the effectiveness of PL, ATT, AT and KD, respectively.

5.6. In-depth Analysis

In this subsection, we compare Prompt-KD with other class-incremental event detection methods to verify its effectiveness on the method level. Subsequently, we conduct experiments to demonstrate the contributions of ATT, AT and PL, which are essential to the performance of Prompt-KD.

5.6.1. Comparison between Prompt-KD and Class-Incremental Event Detection Methods

To further demonstrate the effectiveness of Prompt-KD, we compare it with two class-incremental event detection methods, i.e., EMP (Liu et al., 2022b) and KDR (Yu et al., 2021). EMP is the state-of-the-art method on MAVEN for class-incremental event detection, and KDR is the second state-of-the-art method. The experimental results of 10-way 1-shot tasks on MAVEN are shown in Table 5. As we can see, EMP and KDR achieve worse performance than Prompt-KD, even worse than PA-CRF-CIL. This may be due to that these methods cannot automatically adapt to the few-shot scenarios.

5.6.2. Visualization Analysis of the Attention Mechanism

To further demonstrate the effectiveness of the attention mechanism, we draw a heat map on the 5-way 1-shot tasks on FewEvent and MAVEN to

Method	Dataset: MAVEN					
	Learning Sessions					Average
	1	2	3	4	5	F1
KDR	17.99	15.02	10.56	9.07	6.37	11.80
EMP	18.09	16.03	10.93	8.94	6.43	12.08
PA-CRF-CIL	26.85	21.17	16.67	14.47	11.69	26.64
Prompt-KD	28.40	24.07	19.78	18.23	16.95	30.88

Table 5: Comparison results with class-incremental event detection methods.

Method	#Instances	Dataset: FewEvent					
		Learning Sessions					
		0	1	2	3	4	5
$\mathcal{C}$	50	39	31	25	21	16	12
$\mathcal{D}$	50	39	27	22	17	11	7

Table 6: The case study of the ablated models $\mathcal{C}$ and $\mathcal{D}$

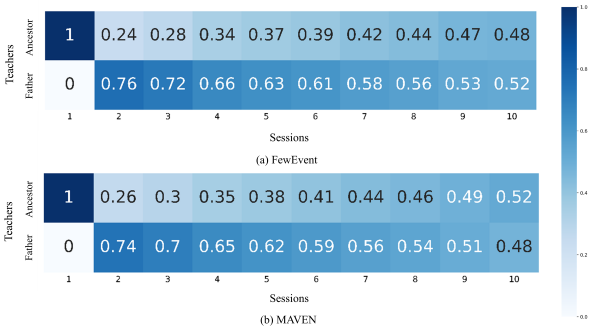


Figure 3: The heat map of different weights of the teacher models on the 5-way 1-shot tasks on FewEvent (a) and MAVEN (b).

visualize the different weights of the ancestor and father teacher models. As shown in Figure 3, the weight of the father teacher model far exceeds that of the ancestor teacher model in the first few learning sessions. However, with new classes arriving constantly, the weight of the ancestor teacher model gradually increases on both two datasets and even exceeds that of the father teacher model on MAVEN. This situation is in line with the intuitive cognition that, the model forgets more knowledge about base classes as time goes by. Therefore, the model needs to assign higher weight to the ancestor teacher model to review the base knowledge.

5.6.3. Case Studies on the Ancestor Teacher Model

To illustrate the contributions of the ancestor teacher model, we conduct experiments on the 10-way 1-shot tasks on FewEvent between the ablated model $\mathcal{C}$ with AT and the ablated model $\mathcal{D}$ without AT. We choose 50 instances (1-shot for 50 base classes) and count how many of them are correctly predicted in the subsequent learning sessions. As shown in Table 6, as the learning session goes by, the model $\mathcal{C}$ predicts more correct instances than the model $\mathcal{D}$. It indicates that the model with AT is less likely to forget the base knowledge, demonstrating the effectiveness of AT in alleviating the old knowledge forgetting problem.

Specifically, we choose two instances of classes *Education.Education* and *Contact.E-Mail* from $D^{(0)}$ of FewEvent and present their prediction results of the models $\mathcal{C}$ and $\mathcal{D}$. As shown in Table 7, the ablated model $\mathcal{C}$ correctly makes predictions on both two instances and in all learning sessions. Nevertheless, the model $\mathcal{D}$ provides wrong answers of the first instance in Sessions 3-5 and of the second one in Sessions 2-5, respectively. This may be because that the model $\mathcal{D}$ puts more attention on the new classes and thus begins to forget base classes after 2-3 learning sessions. The above phenomena can demonstrate that, reusing the ancestor teacher model constantly can effectively overcome the forgetting problem about base knowledge.

5.6.4. Experimental Analysis on Prompt Learning

To verify the effectiveness of prompt learning for coping with the few-shot scenarios and alleviating the corresponding new class overfitting problem, we conduct experiments on the 10-way 1-shot tasks on FewEvent comparing the F1 difference of the ablated method $\mathcal{A}$ with prompt learning and the ablated method $\mathcal{B}$ without prompt learning on the support set and the query set. The experimental results are shown in Table 8, where we can see that the difference of support set and query set decreases and the performance on the query set increases of the model $\mathcal{A}$ comparing with the model $\mathcal{B}$. It indicates that prompt learning can alleviate the new class overfitting problem.

6. Conclusions and Future Work

In this paper, we proposed a new task, i.e., CIFSED, and a knowledge distillation and prompt learning based method, called Prompt-KD, for it. Specifically, to overcome the old knowledge forgetting problem, Prompt-KD develops an attention based multi-teacher knowledge distillation framework,

Instances	Method	Learning Sessions					
		0	1	2	3	4	5
Denis Rancourt is a former professor (*B-Education.Education*) of physics at the University of Ottawa.	$\mathcal{C}$	✓	✓	✓	✓	✓	✓
	$\mathcal{D}$	✓	✓	✓	×	×	×
He says that 20 % of the people who get that card send (*B-Contact.E-mail*) him an e-mail.	$\mathcal{C}$	✓	✓	✓	✓	✓	✓
	$\mathcal{D}$	✓	✓	×	×	×	×

Table 7: The case study of the ablated models $\mathcal{C}$ and $\mathcal{D}$ on the 10-way 1-shot tasks on FewEvent. The blue words denote the ground truth labels, the green check marks indicate that the model provides correct answers, whilst the red cross marks suggest that the model makes wrong predictions.

Dataset: FewEvent							
Method	Dataset	Learning Sessions					Average
		1	2	3	4	5	F1
$\mathcal{A}$	Support Set	67.25	58.41	51.33	43.22	40.78	52.19
$\mathcal{A}$	Query Set	55.48	47.34	40.33	33.79	31.51	41.69
$\mathcal{B}$	Support Set	67.03	58.10	51.07	42.96	40.27	51.88
$\mathcal{B}$	Query Set	54.36	46.12	38.02	30.19	27.22	39.29

Table 8: The F1 scores (%) of the models $\mathcal{A}$ and $\mathcal{B}$ on the support and query set.

where the ancestor teacher model pre-trained on base classes is reused in all learning sessions. Moreover, to cope with the few-shot scenario and alleviate the corresponding new class overfitting problem, Prompt-KD is equipped with a prompt learning mechanism. Extensive experiments on two benchmark datasets, i.e., FewEvent and MAVEN, demonstrate the superior performance of Prompt-KD.

7. Acknowledgments

The work is supported by the National Key Research and Development Project of China, the GFKJ Innovation Project, the Beijing Academy of Artificial intelligence under grant BAAI2019ZD0306, the KGJ Project under grant JCKY2022130C039, and the Lenovo-CAS Joint Lab Youth Scientist Project. We appreciate anonymous reviewers for their insightful comments and suggestions.

8. Bibliographical References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Pengfei Cao, Yubo Chen, Jun Zhao, and Taifeng Wang. 2020. Incremental event detection via knowledge consolidation networks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 707–717.

Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Harandi. 2021. Semantic-aware knowledge distillation for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2534–2543.

Xin Cong, Shiyao Cui, Bowen Yu, Tingwen Liu, Wang Yubin, and Bin Wang. 2021. Few-shot event detection with prototypical amortized conditional random field. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 28–40.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using bart. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845.

Shumin Deng, Ningyu Zhang, Jiaojian Kang, Yichi Zhang, Wei Zhang, and Huajun Chen. 2020. Meta-learning with dynamic-memory-based prototypical network for few-shot event detection. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 151–159.

Ning Ding, Yulin Chen, Xu Han, Guangwei Xu, Pengjun Xie, Hai-Tao Zheng, Zhiyuan Liu, Juanzi Li, and Hong-Gee Kim. 2021. Prompt-learning for fine-grained entity typing. *arXiv preprint arXiv:2108.10604*.

George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. 2004. The automatic content extraction (ace) program-tasks, data,

and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon.

Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. 2021. Few-shot class-incremental learning via relation knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 1255–1263.

G David Forney. 1973. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278.

Jonathan Gordon, John Bronskill, Matthias Bauer, Richard E Turner, Jan Stühmer, and Sebastian Nowozin. 2019. Meta-learning probabilistic inference for prediction. In *International Conference on Learning Representations*.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2022. Ptr: Prompt tuning with rules for text classification. *AI Open*, 3:182–192.

Heng Ji and Ralph Grishman. 2011. Knowledge base population: Successful approaches and challenges. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1148–1158.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Steinar Laenen and Luca Bertinetto. 2021. On episodes, prototypical networks, and few-shot learning. *Advances in Neural Information Processing Systems*, 34:24581–24592.

Sha Li, Liyuan Liu, Yiqing Xie, Heng Ji, and Jiawei Han. 2022. Piled: An identify-and-localize framework for few-shot event detection. *arXiv e-prints*, pages arXiv–2202.

Andy T Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. 2022a. Qaner: Prompting question answering models for few-shot named entity recognition. *arXiv e-prints*, pages arXiv–2203.

Minqian Liu, Shiyu Chang, and Lifu Huang. 2022b. Incremental prompting: Episodic memory prompt for lifelong event detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2157–2165.

Xiaoyu Tao, Xinyuan Chang, Xiaopeng Hong, Xing Wei, and Yihong Gong. 2020a. Topology-preserving class-incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX 16*, pages 254–270. Springer.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. 2020b. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12183–12192.

Rui Wang, Tong Yu, Handong Zhao, Sungchul Kim, Subrata Mitra, Ruiyi Zhang, and Ricardo Henao. 2022. Few-shot class-incremental learning for named entity recognition. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 571–582.

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.

Pengfei Yu, Heng Ji, and Prem Natarajan. 2021. Lifelong event detection with knowledge transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5278–5290.

Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. 2021. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12455–12464.

Kailin Zhao, Xiaolong Jin, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2022. Knowledge-enhanced self-supervised prototypical network for few-shot event detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6266–6275.