Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning

Conference Paper · January 2023

DOI: 10.18653/v1/2023.acl-long.440

5 authors, including:

Zhenrui Yue
University of Illinois Urbana-Champaign
57 PUBLICATIONS **479** CITATIONS
SEE PROFILE

Mengfei Lan
University of Illinois Urbana-Champaign
10 PUBLICATIONS **19** CITATIONS
SEE PROFILE

Dong Wang
University of Illinois Urbana-Champaign
278 PUBLICATIONS **5,695** CITATIONS
SEE PROFILE

Zero- and Few-Shot Event Detection via Prompt-Based Meta Learning

Zhenrui Yue **Huimin Zeng** **Mengfei Lan** **Heng Ji** **Dong Wang**
University of Illinois Urbana-Champaign
{zhenrui3, huiminz3, mlan3, hengji, dwang24}@illinois.edu

Abstract

With emerging online topics as a source for numerous new events, detecting unseen / rare event types presents an elusive challenge for existing event detection methods, where only limited data access is provided for training. To address the data scarcity problem in event detection, we propose MetaEvent, a meta learning-based framework for zero- and few-shot event detection. Specifically, we sample training tasks from existing event types and perform meta training to search for optimal parameters that quickly adapt to unseen tasks. In our framework, we propose to use the cloze-based prompt and a trigger-aware soft verbalizer to efficiently project output to unseen event types. Moreover, we design a contrastive meta objective based on maximum mean discrepancy (MMD) to learn class-separating features. As such, the proposed MetaEvent can perform zero-shot event detection by mapping features to event types without any prior knowledge. In our experiments, we demonstrate the effectiveness of MetaEvent in both zero-shot and few-shot scenarios, where the proposed method achieves state-of-the-art performance in extensive experiments on benchmark datasets FewEvent and MAVEN.

1 Introduction

Event detection tasks have experienced significant improvements thanks to the recent efforts in developing language-based methods (Lu et al., 2021; Pouran Ben Veyseh et al., 2021). One of such methods is pretrained large language models, which can be fine-tuned for detecting events upon input context (Liu et al., 2019; Cong et al., 2021). However, the detection of unseen or rare events remains a challenge for existing methods, as large amounts of annotated data are required for training in a supervised fashion (Shang et al., 2022; Zhang et al., 2022c). For instance, existing models often fail to detect unseen event types due to the lack of domain knowledge, as demonstrated in Figure 1.

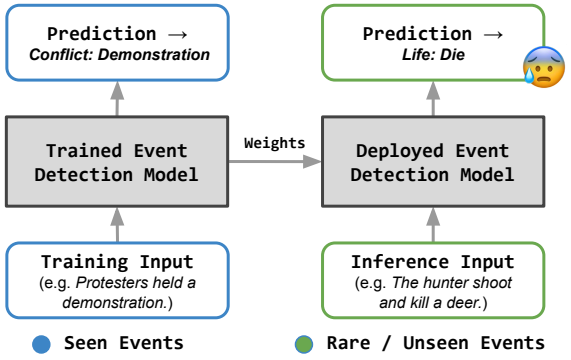

Figure 1: Existing event detection model fails to detect rare / unseen events (green input) upon deployment.

To detect unseen event types (i.e., zero-shot learning), existing methods leverage external knowledge or contrastive learning to build class-separating features (Lyu et al., 2021; Zhang et al., 2022a,b). Similarly, it is possible to leverage limited annotated examples (i.e., few-shot learning) for event detection. For instance, prototypical features can be constructed to match event types upon inference (Deng et al., 2020; Cong et al., 2021). Prompt-based methods align with the pretraining objective of language models to improve detection performance (Li et al., 2022a,b). Specifically, a cloze-based prompt template (e.g., `A <mask> event`) is incorporated as part of the input, and the prediction can be obtained by decoding the `<mask>` prediction. As such, prompt-based methods exploit the masked language modeling (MLM) pretraining task by constructing similar input examples.

Nevertheless, no existing method is designed for both zero-shot and few-shot event detection, as it is a non-trivial problem to combine both settings under a unified framework. Previous efforts either address the zero-shot or the few-shot setting, yet detecting both unseen and rare events can be helpful in various scenarios (e.g., detecting emergency events online), which renders current approaches less effective in real-world applications. Additionally, event detection comprises of two subtasks:

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7928–7943

trigger identification and classification. Some methods provide an incomplete solution by solely performing the classification task, while many other approaches execute both tasks at the cost of reduced performance (Schick and Schütze, 2021; Li et al., 2022b). This is because existing methods heavily rely on trigger words for classification, which causes drastic performance drops in the case of trigger mismatch or ambiguity (Ding et al., 2019).

In this paper, we propose a meta learning framework MetaEvent for zero- and few-shot event detection. We consider both settings and optimize language models to identify unseen / rare events via a unified meta learning framework. That is, given zero / limited number of annotated examples per event type, our objective is to maximize the model performance in detecting such events. To this end, we develop a solution to integrate the trigger identification and classification subtasks for efficient forward passing in meta training. Moreover, we design a trigger-aware soft verbalizer to identify event types in our prompt-based event detection model. For optimization, we propose a meta objective function based on contrastive loss to learn generalizable and class-separating event features. In training, the model is first updated and evaluated upon the sampled zero- or few-shot tasks, then the meta loss is computed to derive gradients w.r.t. the initial parameters, such that the updated model learns to generalize to the target tasks even without labeled examples. In other words, MetaEvent learns from seen tasks, yet with the objective to generalize in rare and unseen scenarios. Therefore, the resulting model can optimally adapt to the target task upon deployment. We demonstrate the effectiveness of MetaEvent by evaluating zero- and few-shot event detection tasks on benchmark datasets, where MetaEvent consistently outperforms state-of-the-art methods with considerable improvements.

We summarize our contributions as follows[1]:

1. To the best of our knowledge, we are the first to propose a unified meta learning framework for both zero- and few-shot event detection. MetaEvent is designed to exploit prompt tuning and contrastive learning for quick adaptation to unseen tasks.

2. We propose an integrated trigger-aware model in MetaEvent for efficient meta training. In particular, our trigger-aware soft verbalizer leverages both the prompt output and attentive trigger features to identify event types.

3. We design a novel contrastive loss as the meta objective of MetaEvent. Our contrastive loss encourages class-separating and generalizable features to improve event matching in both zero- and few-shot event detection.

4. We demonstrate the effectiveness of MetaEvent with extensive experiments, where MetaEvent outperforms state-of-the-art baseline methods with considerable improvements in both zero- and few-shot event detection.

[1]We adopt publicly available datasets in the experiments and release our implementation at https://github.com/Yueeeeeeee/MetaEvent.

2 Related Work

2.1 Event Detection

Event detection refers to the task of classifying event types upon input text. While event detection has achieved progress under the supervised training paradigm (Ji and Grishman, 2008; Lin et al., 2020; Wadden et al., 2019; Liu et al., 2020; Du and Cardie, 2020; Lu et al., 2021; Liu et al., 2022), zero- and few-shot classification remains a challenge due to the lack of prior knowledge and annotated examples. For the zero-shot setting, relevant literature focuses on predefined event knowledge or heuristics to classify unseen events (Huang et al., 2018; Huang and Ji, 2020; Lyu et al., 2021; Zhang et al., 2021b; Yu et al., 2022; Yu and Ji, 2023; Zhan et al., 2023). Similarly, zero-shot contrastive learning requires unlabeled examples in training to learn class-separating features (Zhang et al., 2022b). Under the few-shot setting, prototypical networks and prompt-based tuning improve detection performance via prototype matching and alignment to language pretraining (Deng et al., 2020; Cong et al., 2021; Schick and Schütze, 2021; Lai et al., 2021; Li et al., 2022b). Overall, existing approaches focus on either zero- or few-shot event detection without considering a unified framework for both settings. Moreover, current zero-shot methods require additional resources for training, making such methods less realistic in real-world event detection. As such, we propose a meta learning framework MetaEvent for both zero- and few-shot event detection, where neither prior knowledge nor unlabeled examples are required to detect unseen events.

2.2 Prompt Learning

Prompt learning uses a predefined template with slots (i.e., `A <mask> event`) to instruct language models on the desired task, where the slot prediction is used to derive the final output (Brown et al., 2020; Liu et al., 2021). By leveraging the pretraining objective and designing prompt templates, pretrained large language models can be adapted for zero- or few-shot downstream tasks (Houlsby et al., 2019; Raffel et al., 2020). Soft and multi-task prompts further improve the zero- and few-shot performance of language models on unseen tasks (Lester et al., 2021; Sanh et al., 2021). Cloze-based prompts are proposed for the event detection task, where the predicted output can be mapped to the event types using verbalizer or mapping heuristics (Schick and Schütze, 2021; Li et al., 2022b; Zhang et al., 2022b). Nevertheless, previous prompt methods adopt the inefficient two-step paradigm (i.e., trigger identification and classification) and are not designed for both zero- and few-shot event detection. Therefore, we integrate both steps with a trigger-aware soft verbalizer for efficient forward passing in meta training. Moreover, we consider both zero- and few-shot scenarios with our prompt-based meta learning framework MetaEvent. By leveraging the proposed components, our approach demonstrates considerable improvements compared to existing methods.

2.3 Meta Learning

Meta learning (i.e., learning to learn) has demonstrated superiority in few-shot learning (Finn et al., 2017; Nichol et al., 2018; Rajeswaran et al., 2019). Existing methods learn class-wise features or prototypes in the metric space for quick adaptation to few-shot tasks (Vinyals et al., 2016; Snell et al., 2017; Sung et al., 2018). Model-agnostic meta learning (MAML) leverages the second-order optimization to find the optimal initial parameters for a new task (Finn et al., 2017). Approximation methods of second-order MAML demonstrates comparable performance while requiring significantly reduced computational resources (Finn et al., 2017; Nichol et al., 2018; Rajeswaran et al., 2019). Meta learning has also been applied to tasks like online learning, domain adaptation and multi-task learning (Finn et al., 2019; Li and Hospedales, 2020; Wang et al., 2021a). For event detection, meta learning is proposed to improve the performance on small-size data via memory-based prototypical networks and external knowledge (Deng et al., 2020; Shen et al., 2021).

To the best of our knowledge, meta learning-based methods for both zero- and few-shot event detection is not studied in current literature. However, detecting unseen and rare events is necessary in many applications. An example can be detecting emergency events on social media, where both unseen and rare events can be present (e.g., pandemic, safety alert etc.). Therefore, we propose MetaEvent: a meta learning framework for zero- and few-shot event detection. MetaEvent exploits seen events via trigger-aware prompting and a carefully designed meta objective, and thereby improving the zero- and few-shot performance with class-separating and generalizable features.

3 Preliminary

We consider the following event detection problem setup, where N-*way* K-*shot* examples are available for training each task (K is 0 for zero-shot setting). Our goal is to train a model $\boldsymbol{f}$ that maximizes the performance in unseen tasks.

Problem: Our research focuses on zero- and few-shot event detection based on a collection of tasks $\{\boldsymbol{\mathcal{T}}_i\}_{i=1}^{M}$. For each task $\boldsymbol{\mathcal{T}}_i$, a N-way K-shot training set and a held-out evaluation set are provided (i.e., $\boldsymbol{\mathcal{D}}_i^{\text{train}}, \boldsymbol{\mathcal{D}}_i^{\text{test}} \in \boldsymbol{\mathcal{T}}_i$). The training of MetaEvent is two-fold: (1) an initial model is updated using the training sets in each sampled task to achieve local convergence (i.e., inner-loop optimization); (2) the updated models are used to compute a meta loss on the corresponding evaluation sets, followed by deriving the gradients w.r.t. the initial model using our meta learning algorithm (i.e., outer-loop optimization). In particular, the input for each task $\boldsymbol{\mathcal{T}}_i$ include:

- *Training set*: $\boldsymbol{\mathcal{D}}_i^{\text{train}}$ contains K examples for each of the N classes. An example comprises of context $\boldsymbol{x}_c^{(j)}$, trigger $\boldsymbol{x}_c^{(j)}$ and label $y^{(j)}$ (i.e., $\boldsymbol{\mathcal{D}}_i^{\text{train}} = \{(\boldsymbol{x}_c^{(j)}, \boldsymbol{x}_t^{(j)}, y^{(j)})\}_{j=1}^{N*K}$). In the few-shot setting, $\boldsymbol{\mathcal{D}}_i^{\text{train}}$ is used to tune the model, while $\boldsymbol{\mathcal{D}}_i^{\text{train}}$ is an empty set for zero-shot setting. Training set is also known as support set.

- *Evaluation set*: Similarly, a held-out evaluation set $\boldsymbol{\mathcal{D}}_i^{\text{test}}$ from the same input & label space is used to compute the meta loss in training. Upon deployment, the model can be updated with $\boldsymbol{\mathcal{D}}_i^{\text{train}}$ and should perform well on $\boldsymbol{\mathcal{D}}_i^{\text{test}}$. Evaluation set is often referred to as query set.

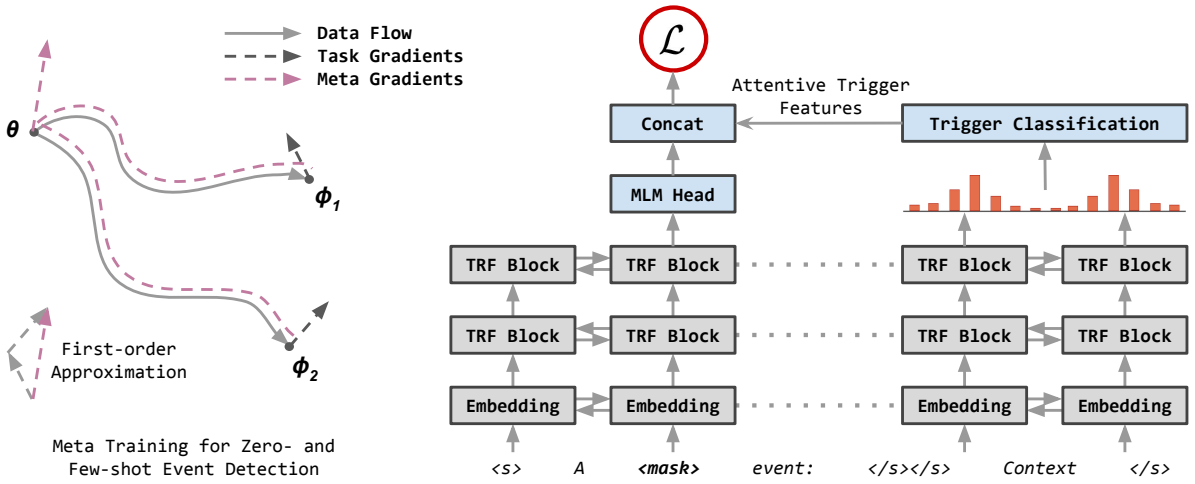

Figure 2: The proposed MetaEvent. The left subfigure illustrates the optimization process w.r.t. the initial parameter set $\boldsymbol{\theta}$ with meta learning, and the right subfigure describes the proposed event detection model in MetaEvent.

Pipeline: We are interested in learning an encoder model $\boldsymbol{f}$ parameterized by $\boldsymbol{\theta}$. Conventional event detection methods compute trigger as an intermediate variable, followed by the event type classification. As such, we formulate $\boldsymbol{f_\theta}$ with contexts as input and event features as output, followed by some classifier CLF to map the features to the desired event type (i.e., $y = \texttt{CLF}(\boldsymbol{f_\theta}(\boldsymbol{x}_c))$). Our goal is to find the optimal parameter set $\boldsymbol{\theta}$ that quickly adapts to an unseen task $\mathcal{T}_i$ using $\mathcal{D}_i^{\text{train}}$ and maximizes the performance on evaluation set $\mathcal{D}_i^{\text{test}}$. Mathematically, this is formulated as optimization of $\boldsymbol{\theta}$ over a collection of M evaluation tasks:

$$\min_{\boldsymbol{\theta}} \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}(\mathcal{A}lg(\boldsymbol{\theta}, \mathcal{D}_i^{\text{train}}), \mathcal{D}_i^{\text{test}}), \quad (1)$$

where $\mathcal{L}$ represents the loss and $\mathcal{A}lg$ represents the gradient descent optimization algorithm.

4 Methodology

4.1 Model Design

To efficiently perform prompt-based meta training for event detection, we design a one-step model that integrates the trigger identification and classification stages. This is because a two-step approach (e.g., P4E (Li et al., 2022b)) requires extensive computational resources to obtain the gradients in the inner- and outer-loop optimization in MetaEvent. Consequently, we design an efficient model (as illustrated in Figure 2) that integrates attentive trigger features to avoid additional forward passes.

Different from existing trigger-aware methods (Ding et al., 2019), the proposed model innovatively uses both attentive trigger features and prompt output to predict the event types. The attentive trigger features $\boldsymbol{t}$ can be computed using an integrated trigger classifier and attention weights from the pretrained language model. Specifically, a trigger classifier is trained upon each token features to perform binary classification (i.e., whether input token is trigger token). In inference, the classifier predicts the probabilities $\boldsymbol{p}$ of input tokens being classified as trigger words. To better estimate the importance of predicted trigger tokens, we design an attentive reweighting strategy to select informative trigger features from the input context. The idea behind our attentive reweighting strategy is to leverage attention scores from the model to select more relevant features. The attention scores reveal different importance weights of the context tokens and thus, can be used to compute 'soft' trigger features based on the semantics. Formally, our attentive reweighting strategy computes weights $\boldsymbol{w} \in \boldsymbol{R}^{L_c}$ using trigger probabilities $\boldsymbol{p}$ and attention scores $A \in \boldsymbol{R}^{H \times L_c \times L_c}$ of the context span from the last transformer layer. The weight of the i-th token w_i in $\boldsymbol{w}$ is computed via

$$w_i = \sigma\left(\boldsymbol{p} \odot \frac{1}{H} \sum_{j}^{H} \left(\sum_{k}^{L_c} A_{j,k} \right) \right)_i, \quad (2)$$

where H is the number of attention heads, L_c is the context length, $\odot$ and σ denote elementwise product and softmax functions. Base on input $\boldsymbol{x}_c$ and $\boldsymbol{w}$, the attentive trigger features $\boldsymbol{t}$ can be computed as the weighted sum of token features, i.e.,

$$\boldsymbol{t} = \sum_{i=1}^{L_c} w_i \boldsymbol{f_\theta}(\boldsymbol{x}_c)_i. \quad (3)$$

For the event classification, we design a prompt-based paradigm using a predefined prompt and a trigger-aware soft verbalizer. Specifically, we preprocess the input context by prepending the prompt '`A <mask> event`' to transform the prediction into a masked language modeling (MLM) task. The pretrained encoder model and MLM head fill the `<mask>` position with a probability distribution $\boldsymbol{v}$ over all tokens. Then, our trigger-aware soft verbalizer maps the predicted distribution $\boldsymbol{v}$ to an output event type. Unlike predefined verbalizer functions (Li et al., 2022b), we design a learnable verbalizer based on MLM predictions $\boldsymbol{v}$ and attentive trigger features $\boldsymbol{t}$. For the N-way few-shot setting, the trigger-aware soft verbalizer with weights $\boldsymbol{W} \in \boldsymbol{R}^{(|\boldsymbol{v}|+|\boldsymbol{t}|)\times N}$ and bias $\boldsymbol{b} \in \boldsymbol{R}^{N}$ predicts the output label with GELU activation via

$$\hat{y} = \arg\max(\mathrm{GELU}([\boldsymbol{v}; \boldsymbol{t}])\boldsymbol{W} + \boldsymbol{b}). \quad (4)$$

For zero-shot event detection, we use the concatenated features $[\boldsymbol{v}; \boldsymbol{t}]$ to project input to unseen event types via the Hungarian algorithm.

4.2 Meta Training

Provided with the training and evaluation sets from sampled tasks, we present the formulation of our MetaEvent and our methods for the zero- and few-shot training. The designed framework leverages meta training to search for optimal parameters $\boldsymbol{\theta}$. Once trained, the event detection model quickly adapts to unseen tasks even without examples (Finn et al., 2017; Yue et al., 2023).

Given a set of tasks $\{\boldsymbol{\mathcal{T}}_i\}_{i=1}^{M}$ and model $\boldsymbol{f}$ parameterized by $\boldsymbol{\theta}$, MetaEvent aims at minimizing the overall evaluation loss of the tasks (as in Equation (1)). MetaEvent consists of an inner-loop optimization stage (i.e., $\mathcal{A}lg$) and an outer-loop optimization staget that minimizes the overall loss w.r.t. $\boldsymbol{\theta}$. For the inner-loop update, $\mathcal{A}lg$ denotes gradient descent with learning rate α, i.e.:

$$\mathcal{A}lg(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}}) = \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}}) = \boldsymbol{\phi}, \quad (5)$$

we denote the updated parameter set with $\boldsymbol{\phi}$. In the outer-level optimization, we are interested in learning an optimal set $\boldsymbol{\theta}$ that minimizes the meta loss on the evaluation sets. The learning is achieved by differentiating through the inner-loop optimization (i.e., $\mathcal{A}lg$) back to the initial parameter set $\boldsymbol{\theta}$, which requires the computation of second-order gradients or first-order approximation (as shown in Figure 2). Specifically, we derive the gradients w.r.t. $\boldsymbol{\theta}$:

$$\frac{d\mathcal{L}}{d\boldsymbol{\theta}} = \frac{d\boldsymbol{\phi}}{d\boldsymbol{\theta}} \nabla_{\boldsymbol{\phi}} \mathcal{L}(\mathcal{A}lg(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}}), \boldsymbol{\mathcal{D}}^{\text{test}}), \quad (6)$$

notice that $\mathcal{A}lg(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}})$ is equivalent to $\boldsymbol{\phi}$. Component $\nabla_{\boldsymbol{\phi}} \mathcal{L}(\mathcal{A}lg(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}}), \boldsymbol{\mathcal{D}}^{\text{test}})$ refers to first-order gradients w.r.t. the task-specific parameter set $\boldsymbol{\phi}$ (i.e., $\mathcal{L} \rightarrow \boldsymbol{\phi}$). The left component $\frac{d\boldsymbol{\phi}}{d\boldsymbol{\theta}}$ tracks parameter-to-parameter changes from $\boldsymbol{\phi}$ to $\boldsymbol{\theta}$ through $\mathcal{A}lg$ (i.e., $\boldsymbol{\phi} \rightarrow \boldsymbol{\theta}$), which involves the computation of the Hessian matrix. As the estimation of the matrix $\frac{d\boldsymbol{\phi}}{d\boldsymbol{\theta}}$ requires extensive computational resources, we provide both first-order and second-order implementations for MetaEvent.

Zero-Shot MetaEvent: For the zero-shot evaluation, the learned initial parameter set should be directly evaluated on $\boldsymbol{\mathcal{D}}^{\text{test}}$ for an unseen task. Therefore, directly optimizing Equation (1) is not feasible for zero-shot event detection, as $\boldsymbol{\mathcal{D}}^{\text{train}}$ is not provided. For optimization, however, training event types can be used for inner-loop optimization of the model. As such, we sample $\boldsymbol{\mathcal{D}}^{\text{train}}$ and $\boldsymbol{\mathcal{D}}^{\text{test}}$ from *different training tasks* to improve the model generalization on unseen events. Specifically for training task $\boldsymbol{\mathcal{T}}_j$, we optimize

$$\min_{\boldsymbol{\theta}} \mathcal{L}(\mathcal{A}lg(\boldsymbol{\theta}, \boldsymbol{\mathcal{D}}^{\text{train}} \sim \{\boldsymbol{\mathcal{T}}_i\}_{i=1, i\neq j}^{M}), \boldsymbol{\mathcal{D}}_j^{\text{test}}), \quad (7)$$

where $\boldsymbol{\mathcal{D}}^{\text{train}}$ is a disjoint training set sampled from $\{\boldsymbol{\mathcal{T}}_i\}_{i=1, i\neq j}^{M}$. As a result, the model 'learns to adapt' to unseen events by optimizing on different training and evaluation sets. To improve the performance on unseen event types via Hungarian algorithm, we additionally design a contrastive loss term in the meta objective to learn class-separating features, which is introduced in Section 4.3.

Few-Shot MetaEvent: In the few-shot event detection, we directly sample training tasks and optimize the model as in Equation (1). Similar to the zero-shot MetaEvent, the parameters are updated upon the tasks separately (i.e., $\boldsymbol{\phi}$) in each iteration based on the initial parameters $\boldsymbol{\theta}$. Then, the meta loss $\mathcal{L}$ and gradients w.r.t. $\boldsymbol{\theta}$ are computed for each task using the updated $\boldsymbol{\phi}$ and the evaluation sets. In our implementation, we adopt layer- and step-adaptive learning rates for inner-loop optimization and cosine annealing to improve the convergence of MetaEvent (Antoniou et al., 2018).

4.3 Meta Objective

We now introduce our training objective $\mathcal{L}$ for MetaEvent. Based on the proposed model in Sec-

tion 4.1, our loss function contains two classification losses: trigger classification loss $\mathcal{L}_{\text{trigger}}$ and event classification loss $\mathcal{L}_{\text{event}}$ (i.e., negative log likelihood loss). To enlarge the inter-class event discrepancy for improved zero- and few-shot performance, we additionally propose a contrastive loss $\mathcal{L}_{\text{con.}}$ based on the maximum mean discrepancy (MMD) (Gretton et al., 2012; Yue et al., 2021, 2022b,a). In particular, we measure the discrepancy between two different event types by estimating the MMD distance. MMD computes the distance between two event distributions using an arbitrary number of input features drawn from these event types. Mathematically, MMD distance between input features $\boldsymbol{X}$ and $\boldsymbol{Y}$ can be computed as:

$$\begin{aligned}\mathcal{D}(\boldsymbol{X}, \boldsymbol{Y}) = &\frac{1}{|\boldsymbol{X}||\boldsymbol{X}|}\sum_{i=1}^{|\boldsymbol{X}|}\sum_{j=1}^{|\boldsymbol{X}|} k(\psi(\boldsymbol{x}^{(i)}), \psi(\boldsymbol{x}^{(j)})) \\ &+ \frac{1}{|\boldsymbol{Y}||\boldsymbol{Y}|}\sum_{i=1}^{|\boldsymbol{Y}|}\sum_{j=1}^{|\boldsymbol{Y}|} k(\psi(\boldsymbol{y}^{(i)}), \psi(\boldsymbol{y}^{(j)})) \\ &- \frac{2}{|\boldsymbol{X}||\boldsymbol{Y}|}\sum_{i=1}^{|\boldsymbol{X}|}\sum_{j=1}^{|\boldsymbol{Y}|} k(\psi(\boldsymbol{x}^{(i)}), \psi(\boldsymbol{y}^{(j)})),\end{aligned} \tag{8}$$

where k represents the Gaussian kernel and ψ represents the feature mapping function defined by the transformer network (i.e., feature encoder).

Based on the MMD distance, we further compute the inter-class distances for all pairs of event types. Suppose $\boldsymbol{X}_i$ represents the set of trigger-aware event features (i.e., concatenation of $[\boldsymbol{v}; \boldsymbol{t}]$ as in Section 4.1) of the i-th class, the contrastive loss can be formulated for the N-way setting as:

$$\mathcal{L}_{\text{con.}} = -\frac{1}{N(N-1)}\sum_{i=1}^{N}\sum_{j=1, j\neq i}^{N}\mathcal{D}(\boldsymbol{X}_i, \boldsymbol{X}_j), \tag{9}$$

in which we compute $N \times (N-1)$ pairs of inter-class distances, such inter-class distances are maximized (by taking the negative values) in the meta objective function to encourage class-separating features in MetaEvent, and therefore improves both zero- and few-shot performance in event detection.

Overall Objective: We now combine the mentioned terms into a single optimization objective for MetaEvent in Equation (10). In our objective function, $\mathcal{L}_{\text{event}}$ and $\mathcal{L}_{\text{trigger}}$ represent the event and trigger classification loss (i.e., negative log likelihood loss), and $\mathcal{L}_{\text{con.}}$ denotes the contrastive loss based on MMD. The overall objective contains three terms and $\mathcal{L}_{\text{con.}}$ is weighted by a scaling factor λ_c (to be chosen empirically):

$$\mathcal{L} = \mathcal{L}_{\text{event}} + \mathcal{L}_{\text{trigger}} + \lambda_{\text{c}}\mathcal{L}_{\text{con.}}. \tag{10}$$

4.4 Overall Framework

The overall framework of MetaEvent is presented in Figure 2. The right subfigure illustrates the proposed model that integrates attentive trigger features for prompt-based event detection. The left subfigure illustrates the meta training paradigm of MetaEvent, where the initial parameter $\boldsymbol{\theta}$ is updated and evaluated upon sampled tasks using the proposed contrastive loss in Equation (10). Then the gradients w.r.t. $\boldsymbol{\theta}$ can be computed (via second-order optimization or first-order approximation) to update the initial model. Unlike previous works (Schick and Schütze, 2021; Cong et al., 2021; Li et al., 2022b; Zhang et al., 2022b), we design a trigger-aware model for efficient training and inference on event detection tasks. Moreover, we propose a meta learning framework MetaEvent with a fine-grained contrastive objective function for zero- and few-shot event detection, which encourages generalizable and class-separating features across both seen and unseen event types.

5 Experiments

5.1 Settings

Model: Following previous work (Wang et al., 2021b; Li et al., 2022b), we select RoBERTa as the base model in MetaEvent (Liu et al., 2019).

Evaluation: To validate the proposed method, we follow (Chen et al., 2021; Cong et al., 2021; Li et al., 2022b) to split the datasets into train, validation and test sets. For evaluation metrics, we adopt micro F1 score as main performance indicator. For the zero-shot setting, we additionally adopt adjusted mutual information (AMI) and Fowlkes Mallows score (FM) to evaluate clustering performance. See evaluation details in Appendix A.

Datasets and Baselines: To examine MetaEvent performance, we adopt publicly available datasets FewEvent and MAVEN (Deng et al., 2020; Wang et al., 2020) and state-of-the-art baseline methods for comparison. For zero-shot baselines, we adopt SCCL (Zhang et al., 2021a), SS-VQ-VAE (Huang and Ji, 2020), BERT-OCL (Zhang et al., 2022b) and ZEOP (Zhang et al., 2022b). For the few-shot setting, we choose BERT-CRF (Devlin et al., 2019), PA-CRF (Cong et al., 2021), Prompt+QA (Li et al.,

Method	FewEvent			MAVEN		
	F1 ↑	AMI ↑	FM ↑	F1 ↑	AMI ↑	FM ↑
SCCL	0.3184	0.2371	0.2436	0.2424	0.1546	0.1483
SS-VQ-VAE	0.3670	0.3462	0.2758	0.1934	0.1192	0.1838
BERT-OCL	0.3296	0.5326	0.4016	0.1446	0.1915	0.1160
ZEOP	0.4869	0.4065	0.3392	0.2444	0.1274	0.1642
ZEOP*	0.5655	0.5135	0.4360	0.2383	0.1366	0.1484
MetaEvent	**0.6837**$_{\pm 0.0689}$	**0.6884**$_{\pm 0.0315}$	**0.7247**$_{\pm 0.0807}$	**0.3686**$_{\pm 0.0412}$	**0.2352**$_{\pm 0.0521}$	**0.2569**$_{\pm 0.0392}$

Table 1: Zero-Shot event detection results (10-way for both datasets).

2022b) and P4E (Li et al., 2022b) as baselines[2]. Dataset and baseline details are in Appendix A

Implementation: We use the `roberta-base` variant in our implementation, our default model is trained with AdamW optimizer with zero weight decay and cosine annealing for meta learning rate of $1e-5$. For inner-loop optimization, we use layer- and step-adaptive learning rates with an intial learning rate of $1e-3$, where the model is updated 50 times in each task. We select the best model on the validation set for evaluation on the test set. For baseline methods, we follow the reported training procedure and hyperparameter settings from the original works unless otherwise suggested. More implementation and experiment details are provided in Appendix A.

5.2 Zero-Shot Results

We first report zero-shot results on all datasets in Table 1, which is divided into two parts by the used datasets. We perform 10-way event detection on unseen tasks from the disjoint test sets and report the evaluation results, the best results are marked bold, the second best results are underlined. We observe: (1) the zero-shot performance on FewEvent is comparatively higher than MAVEN for both baselines and MetaEvent, possibly due to the increased coverage of event domains in MAVEN. (2) MetaEvent performs the best on both datasets, in particular, MetaEvent achieves 35.9% average improvements on F1 compared to the second best-performing method. (3) Despite lower performance on MAVEN, MetaEvent outperforms all baseline methods with up to 50.8% improvements on F1, suggesting the effectiveness of the proposed meta training algorithm in detecting unseen events.

We further study the effectiveness of the proposed contrastive loss quantitatively in zero-shot

[2]We additionally select the adapted ZEOP*, PA-CRF* and P4E* as improved variants of the baselines, see Appendix A.2.

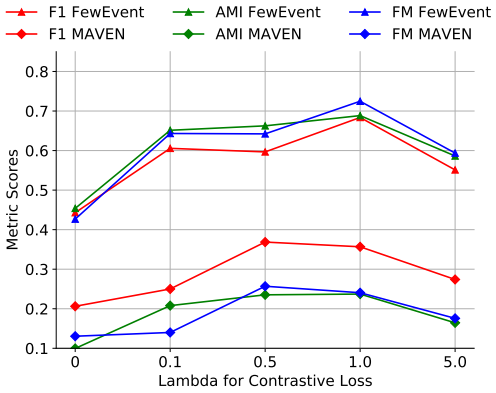

Figure 3: Sensitivity analysis of λ_c.

Figure 4: Feature visualization of MetaEvent w/o (left subfigure) and w/ (right subfigure) contrastive learning.

experiments by varying scaling factor λ_c. Specifically, we choose λ_c from 0 to 5 and evaluate the performance changes on both datasets. The results are visually presented in Figure 3, from which we observe: (1) by applying the contrastive loss (i.e., $\lambda_c \neq 0$) in the meta objective, the zero-shot performance consistently improves regardless of the choice of λ_c; (2) despite huge improvements, carefully chosen λ_c is required for the best zero-shot performance (up to $\sim 100\%$ increases across all metrics). Overall, the contrastive objective proposed in MetaEvent is particularly effective in learning class-separating features, and thereby improves the zero-shot event detection performance.

We additionally present qualitative analysis of our zero-shot results by comparing the feature vi-

Method	FewEvent		MAVEN	
	F1 (K=5) ↑	F1 (K=10) ↑	F1 (K=5) ↑	F1 (K=10) ↑
BERT-CRF	0.4406	0.6673	0.4814	0.6468
PA-CRF	0.5848	0.6164	0.4257	0.4918
PA-CRF*	0.6364	0.7069	0.5316	0.6562
Prompt + QA	0.6523	0.6750	0.4786	0.6543
P4E	0.8198	0.8550	0.6064	0.6951
P4E*	$0.9070_{\pm 0.0220}$	$0.9270_{\pm 0.0110}$	$0.6390_{\pm 0.0090}$	$0.7260_{\pm 0.0150}$
MetaEvent	$\mathbf{0.9318}_{\pm 0.0216}$	$\mathbf{0.9576}_{\pm 0.0052}$	$\mathbf{0.9306}_{\pm 0.0026}$	$\mathbf{0.9486}_{\pm 0.0003}$

Table 2: Few-shot event detection results (10-way for FewEvent and 45-way for MAVEN).

Method	F1 (K=5) ↑	F1 (K=10) ↑
MetaEvent	0.9318	0.9576
w/o Trigger	0.9170	0.9367
w/o Verbalizer	0.8117	0.8516
w/o Meta Learner	0.6257	0.6390

Table 3: Ablation study of MetaEvent.

Method	F1 (K=5) ↑	F1 (K=10) ↑
Prompt A	0.9318	0.9576
Prompt B	0.9363	0.9644
Prompt C	0.9272	0.9527
Prompt D	0.9236	0.9592

Table 4: Analysis of different prompt design.

sualization (via T-SNE) with and without the proposed contrastive loss. We use color to represent different event types and present the visualization in Figure 4. With the contrastive loss (right subfigure), the model generates class-separating features on unseen events compared to MetaEvent trained without contrastive loss (left subfigure). Examples of the same event type are also more aggregated, showing improved clustering results, which can be combined with trigger predictions for identifying unseen event types. In sum, the proposed contrastive loss demonstrates effectiveness in zero-shot settings and consistently outperforms baselines.

5.3 Few-Shot Results

To examine the effectiveness of our method for both zero- and few-shot scenarios, we also perform few-shot experiments on both datasets. The 5-shot and 10-shot event detection experiment results are presented in Table 2, where all evaluation event types are used (10-way for FewEvent and 45-way for MAVEN[3]). We observe: (1) all baseline methods and MetaEvent achieve improved performance in few-shot event detection compared to the zero-shot results. For example, MetaEvent achieves 36.3% F1 improvement in 5-shot setting on FewEvent. (2) For MAVEN, the average performance of all baseline methods are comparatively lower than FewEvent, indicating the challenge of few-shot classification with increased number of event types. (3) MetaEvent achieves the best results, outperforming the second best method in F1 by 3.0% (on FewEvent) and 38.1% (on MAVEN) on average across both few-shot settings. Overall, MetaEvent achieves state-of-the-art performance in event detection even only with few examples.

We now perform ablation studies in the few-shot setting to evaluate the effectiveness of the proposed component in MetaEvent. In particular, we remove the proposed attentive trigger features (i.e., trigger), trigger-aware soft verbalizer (i.e., verbalizer) and outer-loop optimization (i.e., meta learner) in sequence to observe the performance changes on FewEvent. The results are reported in Table 3. For all components, we observe performance drops when removed from MetaEvent. In the 5-shot setting, F1 score reduces by 1.6% and 12.9% respetively when removing the attentive trigger features and trigger-aware soft verbalizer. The results suggest that proposed components are effective for improving few-shot event detection.

Finally, we study the influence of prompt designs and report the results on FewEvent in Table 4. In particular, we select from prompt A: '`A <mask> event`', B: '`This text describes a <mask> event`', C: '`This topic is about <mask>`' and D: '`[Event: <mask>]`'. From the results we observe: for 5-shot event detection, prompt A and B

[3]Similar to (Chen et al., 2021), we perform binary classification for each of the event types in MAVEN as multiple event labels may exist on the same input context.

perform the best while prompt B and D achieves better performance with 0.9644 and 0.9592 F1 in 10-shot setting. On average, prompt B outperforms all other prompt designs in the F1 metric, indicating that well-designed instructions may slightly improve few-shot event detection results.

6 Conclusion

In this paper, we design a meta learning framework MetaEvent for zero- and few-shot event detection. MetaEvent proposes to leverage attentive trigger features for efficient inference and predicts via a trigger-aware soft verbalizer. Moreover, the proposed MetaEvent trains the model to search for the optimal parameter set for quick adaptation to unseen event detection tasks. Extensive experiment results demonstrate the effectiveness of MetaEvent by consistently outperforming state-of-the-art methods on benchmark datasets in both zero- and few-shot event detection.

7 Limitations

While the proposed MetaEvent achieves significant improvements in both zero- and few-shot event detection, MetaEvent requires additional computational resources due to the layer- and step-adaptive learning rates and the outer-loop optimization, which may cause increased computational costs for training MetaEvent. Moreover, we have not investigated the benefits of task scheduling techniques and similarity-based meta learning in MetaEvent to fully explore the training event types. Consequently, we plan to study efficient meta learning with advanced training task scheduling for further improvements in zero- and few-shot event detection as future work.

Acknowledgments

This research is supported in part by the National Science Foundation under Grant No. IIS-2202481, CHE-2105005, IIS-2008228, CNS-1845639, CNS-1831669, U.S. DARPA KAIROS Program No. FA8750-19-2-1004 and U.S. DARPA AIDA Program No. FA8750-18-2-0014. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Antreas Antoniou, Harrison Edwards, and Amos Storkey. 2018. How to train your maml. *arXiv preprint arXiv:1810.09502*.

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.

Jiawei Chen, Hongyu Lin, Xianpei Han, and Le Sun. 2021. Honey or poison? solving the trigger curse in few-shot event detection via causal intervention. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8078–8088, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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

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

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

Ning Ding, Ziran Li, Zhiyuan Liu, Haitao Zheng, and Zibo Lin. 2019. Event detection with trigger-aware lattice neural network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 347–356, Hong Kong, China. Association for Computational Linguistics.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. 2019. Online meta-learning. In *International Conference on Machine Learning*, pages 1920–1930. PMLR.

Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. 2012. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 718–724, Online. Association for Computational Linguistics.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare Voss. 2018. Zero-shot transfer learning for event extraction. In *Proc. The 56th Annual Meeting of the Association for Computational Linguistics (ACL2018)*.

Heng Ji and Ralph Grishman. 2008. Refining event extraction through unsupervised cross-document inference. In *In Proceedings of the Annual Meeting of the Association of Computational Linguistics (ACL 2008). Ohio, USA*.

Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Learning prototype representations across few-shot tasks for event detection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5270–5277, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Da Li and Timothy Hospedales. 2020. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer.

Haochen Li, Tong Mo, Hongcheng Fan, Jingkun Wang, Jiaxi Wang, Fuhao Zhang, and Weiping Li. 2022a. KiPT: Knowledge-injected prompt tuning for event detection. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1943–1952, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sha Li, Liyuan Liu, Yiqing Xie, Heng Ji, and Jiawei Han. 2022b. Piled: An identify-and-localize framework for few-shot event detection. *arXiv preprint arXiv:2202.07615*.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *Proc. The 58th Annual Meeting of the Association for Computational Linguistics (ACL2020)*.

Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1641–1651, Online. Association for Computational Linguistics.

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

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

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

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2795–2806, Online. Association for Computational Linguistics.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 322–332, Online. Association for Computational Linguistics.

Alex Nichol, Joshua Achiam, and John Schulman. 2018. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*.

Amir Pouran Ben Veyseh, Viet Lai, Franck Dernoncourt, and Thien Huu Nguyen. 2021. Unleash GPT-2 power for event detection. In *Proceedings of the 59th*

Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 6271–6282, Online. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Aravind Rajeswaran, Chelsea Finn, Sham M Kakade, and Sergey Levine. 2019. Meta-learning with implicit gradients. *Advances in neural information processing systems*, 32.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2021. Multitask prompted training enables zero-shot task generalization. *arXiv preprint arXiv:2110.08207*.

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

Lanyu Shang, Yang Zhang, Christina Youn, and Dong Wang. 2022. Sat-geo: A social sensing based content-only approach to geolocating abnormal traffic events using syntax-based probabilistic learning. *Information Processing & Management*, 59(2):102807.

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

Jake Snell, Kevin Swersky, and Richard Zemel. 2017. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30.

Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. 2018. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. *Advances in neural information processing systems*, 29.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5784–5789, Hong Kong, China. Association for Computational Linguistics.

Haoxiang Wang, Han Zhao, and Bo Li. 2021a. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation. In *International Conference on Machine Learning*, pages 10991–11002. PMLR.

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

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021b. CLEVE: Contrastive Pre-training for Event Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6283–6297, Online. Association for Computational Linguistics.

Pengfei Yu and Heng Ji. 2023. Shorten the long tail for rare entity and event extraction. In *Proc. The 17th Conference of the European Chapter of the Association for Computational Linguistics (EACL2023)*.

Pengfei Yu, Zixuan Zhang, Clare Voss, Jonathan May, and Heng Ji. 2022. Event extractor with only a few examples. In *Proc. NAACL2022 workshop on Deep Learning for Low Resource NLP*.

Zhenrui Yue, Bernhard Kratzwald, and Stefan Feuerriegel. 2021. Contrastive domain adaptation for question answering using limited text corpora. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9575–9593, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022a. Contrastive domain adaptation for early misinformation detection: A case study on covid-19. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pages 2423–2433.

Zhenrui Yue, Huimin Zeng, Ziyi Kou, Lanyu Shang, and Dong Wang. 2022b. Domain adaptation for question answering via question classification. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1776–1790, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Zhenrui Yue, Huimin Zeng, Yang Zhang, Lanyu Shang, and Dong Wang. 2023. Metaadapt: Domain adaptive few-shot misinformation detection via meta learning. *arXiv preprint arXiv:2305.12692*.

Qiusi Zhan, Sha Li, Kathryn Conger, Martha Palmer, Heng Ji, and Jiawei Han. 2023. Glen: General-purpose event detection for thousands of types. In *arXiv*.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. Supporting clustering with contrastive learning. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5419–5430, Online. Association for Computational Linguistics.

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

Hongming Zhang, Wenlin Yao, and Dong Yu. 2022a. Efficient zero-shot event extraction with context-definition alignment. *arXiv preprint arXiv:2211.05156*.

Senhui Zhang, Tao Ji, Wendi Ji, and Xiaoling Wang. 2022b. Zero-shot event detection based on ordered contrastive learning and prompt-based prediction. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 2572–2580, Seattle, United States. Association for Computational Linguistics.

Yang Zhang, Ruohan Zong, Lanyu Shang, Ziyi Kou, and Dong Wang. 2022c. An active one-shot learning approach to recognizing land usage from class-wise sparse satellite imagery in smart urban sensing. *Knowledge-Based Systems*, 249:108997.

A Implementation

A.1 Datasets

We adopt FewEvent and MAVEN for our experiments, the details of the datasets are reported below. **FewEvent** is a dataset designed for few-shot event detection (Deng et al., 2020). We follow the preprocessing of (Cong et al., 2021; Li et al., 2022b) and present the data statistics in Table 5. FewEvent contains 100 event types with three disjoint sets of event classes in training, validation and test sets. The dataset is based on ACE and TAC-KBP with new event types from Freebase and Wikipedia.

	Train	Valid	Test
Classes	80	10	10
Examples	68506	2173	697

Table 5: Dataset statistics of FewEvent.

MAVEN is a large event detection dataset with over 150 event types and over 80k event mentions in total (Wang et al., 2020). We follow the preprocessing of (Chen et al., 2021; Li et al., 2022b) and present the data statistics in Table 6. MAVEN covers an enlarged set of event types with increased examples per class. Unlike FewEvent, the event types in the validation and test sets are overlapping. Since MAVEN provide multi-label examples, we perform binary classification for each of the event types, we additionally sample 10 times the negative examples for both training and evaluation.

	Train	Valid	Test
Classes	125	45	45
Examples	79906	1532	1555

Table 6: Dataset statistics of MAVEN.

A.2 Baseline Methods

We introduce the details of the zero-shot baseline methods, followed by the baseline methods in the few-shot setting. For zero-shot methods that leverage unseen event examples in training (e.g., ZEOP), we adapt the baseline methods by dividing the training set into seen and unseen event types. As such, unseen examples can be sampled from the training set and no examples from the evaluation event types are participated in training.

Supporting Clustering with Contrastive Learning (SCCL) is a clustering-based approach for unsupervised classification. SCCL is used to detect new event types based on unseen event mentions. The contextual feature of trigger tokens are used in our experiments (Zhang et al., 2021a).

Semi-supervised Vector Quantized Variational Autoeocoder (SS-VQ-VAE) leverages variational autoencoder to learn discrete event features. SS-VQ-VAE is trained on seen event types and annotations and can be adapted for zero-shot event detection (Huang and Ji, 2020).

BERT Ordered Contrastive Learning (BERT-OCL) designs an ordered contrastive learning method for clustering unseen event types. The Euclidean distance is used to compute pair-wise distance between examples for reducing intra-class distances and increasing inter-class distances (Devlin et al., 2019; Zhang et al., 2022b).

Zero-Shot Event Detection with Ordered Contrastive Learning (ZEOP & ZEOP*) leverages prompt learning and ordered contrastive loss based on both instance-level and class-level distance for zero-shot event detection. ZEOP first identifies trigger tokens then predicts event types by clustering, while ZEOP* predicts without inference on trigger words (Zhang et al., 2022b).

The following methods are the few-shot baseline methods used in our experiments.

BERT Conditional Random Field (BERT-CRF) uses BERT encoder with a conditional random field classifier used to classify tokens. BERT-CRF can be fine-tuned on event detection tasks with limited examples (Devlin et al., 2019).

Prototypical Amortized Conditional Random Field (PA-CRF & PA-CRF*) improves upon BERT-CRF by estimating transition scores and class uncertainty based on label prototypes and Gaussian distributions. PA-CRF* memories the prototypes and recomputes them in each iteration to achieve improved performance (Cong et al., 2021).

Prompt + Question Answering (Prompt+QA) leverages both prompt-based classification and question answering to: (1) make inference on the event type with predefined prompt and a verbalizer; (2) perform QA task to query the trigger tokens based on the previous classification results (Du and Cardie, 2020; Li et al., 2022b).

Prompt-Guided Event Detection (P4E & P4E*) proposed a prompt-based approach to first identify event types, followed by trigger localization using the previous output. P4E achieves state-of-the-art performance by prompt-tuning pre-

trained language models on event detection tasks. P4E* only performs event type classification without inference on trigger words (Li et al., 2022b).

A.3 Implementation Details

For our evaluation method, we follow the previous works (Chen et al., 2021; Cong et al., 2021; Li et al., 2022b; Zhang et al., 2022b) and split the datasets into train, validation, and test sets. The validation sets are used for selecting the best model (with validation F1 score) to perform evaluation. For baseline implementation, we follow the reported training procedure and hyperparameter settings from the original works unless otherwise suggested. However, for baseline methods that require unlabeled examples from unseen classes in training (e.g., ZEOP), we modify such methods by sampling event types from the training set as unseen events. As such, the baseline methods can be trained without test event types being participated in the optimization. As a result, we observe slight performance drops compared to the original implementation (Zhang et al., 2022b). For few-shot event detection baseline methods, the results are directly taken from (Li et al., 2022b).

In MetaEvent optimization, the outer-loop learning rates are selected from $[1e-5, 2e-5, 3e-5]$, the initial inner-loop learning rates are selected from $[1e-4, 1e-3]$[4], the learning rate of adaptive learning rate is $1e-4$. MetaEvent uses the AdamW optimizer without cosine annealing learning rate scheduler in meta optimization. For inner-loop, the maximum batch size is 50 and we leverage per-layer per-step adaptive learning rates and perform 50 updates in total (Antoniou et al., 2018). We adopt $[2, 3]$ as the number of tasks, number of iterations are selected from $[250, 500]$ depending on the task and size of the dataset. The model is validated every 25 iterations, the best model in validation is used to evaluate on the test split. For hyperparameter, see sensitivity analysis in Section 5. All reported results are based on first-order meta learning approximation and experiments are performed on multiple NVIDIA A40 GPUs.

B Additional Results

In this section, we present additional results on zero-shot performance of MetaEvent. Specifically, we provide additional clustering metrics for MetaEvent on both datasets, with the results presented in Table 7. We adopt the following clustering metrics: (1) rand score (Rand Score); (2) adjusted rand score (Adjusted Rand); (3) normalized mutual information (Normalized MI) and (4) homegeneity score (Homogeneity Score). Surprisingly, the rand score and adjusted rand score demonstrates significant difference on MAVEN, suggesting disproportionate label distribution in MAVEN. For the rest metrics on both datasets, we observe similar magnitude of performance as reported in Section 5. Moreover, we present additional visualization results on FewEvent below with varying λ_c values. Compared to Figure 4, we observe that cluster aggregation worsens with reduced λ_c values.

Metric	FewEvent	MAVEN
Reported F1	0.6837	0.3686
Rand Score	0.9164	0.8387
Adjusted Rand	0.6780	0.2001
Normalized MI	0.6719	0.3248
Homogeneity Score	0.6779	0.3305

Table 7: Additional zero-shot results of MetaEvent.

Figure 5: Feature visualization ($\lambda_c = 0.1$).

Figure 6: Feature visualization ($\lambda_c = 0.5$).

[4]For inner-loop optimization, the verbalizer weights are initialized with 10 times the base learning rate (i.e., inner-loop learning rate) for faster convergence.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
7 Limitations

☑ A2. Did you discuss any potential risks of your work?
7 Limitations

☑ A3. Do the abstract and introduction summarize the paper's main claims?
7 Limitations

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

B ☑ Did you use or create scientific artifacts?

5 Experiments

☑ B1. Did you cite the creators of artifacts you used?
5 Experiments

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
5 Experiments & A.1 Datasets

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
5 Experiments & A.1 Datasets

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
5 Experiments & A.1 Datasets

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
5 Experiments & A.1 Datasets

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
A.1 Datasets

C ☑ Did you run computational experiments?

5 Experiments

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
5 Experiments & A.3 Implementation Details

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

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
5 Experiments & A.3 Implementation Details

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

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5 Experiments & A.3 Implementation Details

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

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.