Zero-Shot Event Detection Based on Ordered Contrastive Learning and Prompt-Based Prediction

Senhui Zhang[1], **Tao Ji**[1], **Wendi Ji**[1(✉)], **Xiaoling Wang**[1,2]
Shanghai Key Laboratory of Trustworthy Computing,
East China Normal University, Shanghai 200062, China[1]
Shanghai Institute of Intelligent Science and Technology,
Tongji University, Shanghai 200092, China[2]
51194501081@stu.ecnu.edu.cn, taoji.cs@gmail.com
wdji@cs.ecnu.edu.cn, xlwang@cs.ecnu.edu.cn

Abstract

Event detection is a classic natural language processing task. However, the constantly emerging new events make supervised methods not applicable to unseen types. Previous zero-shot event detection methods either require predefined event types as heuristic rules or resort to external semantic analyzing tools. To overcome this weakness, we propose an end-to-end framework named **Z**ero-Shot **E**vent Detection Based on **O**rdered Contrastive Learning and **P**rompt-Based Prediction (ZEOP). By creatively introducing multiple contrastive samples with ordered similarities, the encoder can learn event representations from both instance-level and class-level, which makes the distinctions between different unseen types more significant. Meanwhile, we utilize the prompt-based prediction to identify trigger words without relying on external resources. Experiments demonstrate that our model detects events more effectively and accurately than state-of-the-art methods.

1 Introduction

As a classic NLP task, event detection aims to identify events from natural language text. Most traditional supervised event detection methods (Nguyen and Grishman, 2018; Wadden et al., 2019; Lin et al., 2020) rely on a great number of event-specific annotated texts. However, in practice, obtaining large-scale and high-quality annotated data requires significant expertise and expensive resources. In the real-world scenarios shown in Figure 1, the constantly emerging of new events without annotated samples, making supervised event detection methods no longer applicable.

To solve this challenge, the zero-shot event detection task is proposed to automatically discover and classify new events from unstructured texts in the absence of manual annotation. Following previous works (Zhang et al., 2015; Huang et al., 2018; Huang and Ji, 2020; Wang et al., 2021), we denote the known types as *seen* types and the new types as *unseen* types. "Unseen" means that the event labels of samples are not visible to the model.

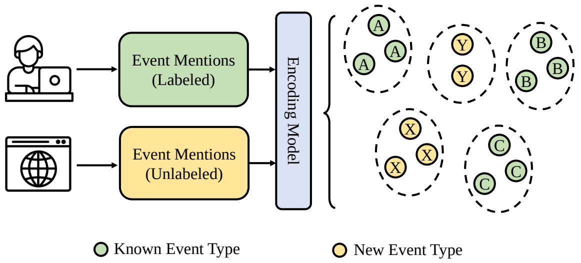

Figure 1: Zero-Shot Event Detection. A training dataset with a few known event types is already annotated manually. The Internet continually produce unlabeled text data every second, which contains a large number of new event types.

Recently, multiple zero-shot event detection methods have been proposed and show better performance than supervised methods on zero-shot tasks. However, they (Huang et al., 2018; Zhang et al., 2021b; Lyu et al., 2021; Huang and Ji, 2020) all require predefined event types as heuristic rules or external semantic analyzing tools. For example, event names are used to query trigger words by semantical similarity, or the part-of-speech tagging tools are used to find nouns and verbs in the text as candidate trigger words. In these ways, human effort and external resources are still necessary when detecting new event types. There is also a problem of error accumulation across the tools and models.

To overcome the above weakness, this paper proposes an end-to-end model named Zero-Shot Event Detection Based on Ordered Contrastive Learning and Prompt-Based Prediction. The main idea is introducing contrastive learning to move away from the dependence on heuristic rules for unseen event detection. As shown in Figure 2, traditional

Findings of the Association for Computational Linguistics: NAACL 2022, pages 2572 - 2580
July 10-15, 2022 ©2022 Association for Computational Linguistics

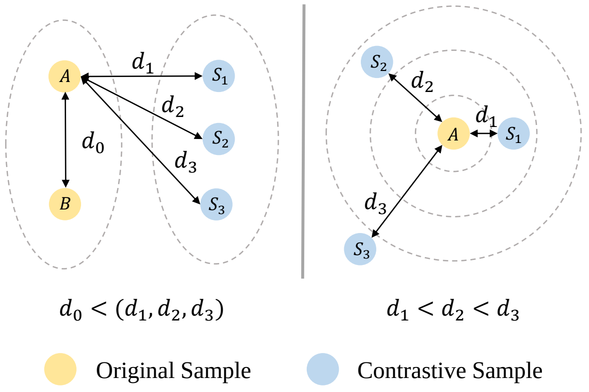

Figure 2: Traditional contrastive learning (left) only distinguish between positive and negative samples, while ordered contrastive learning (right) constructs an ordered sequence of contrastive samples by the similarity to the original sample.

contrastive learning simply divide samples into two opposite classes: positive or negative, while we construct four contrastive samples with different similarities to the original sample. Then, the ordered contrastive learning can draw a stronger distinction between different unseen type events by learning the partial order relationship of different contrastive samples. Meanwhile, in order to discover new event types without relying on heuristic rules, we utilize the prompt-based prediction (Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021a) for trigger words identification, which has been proved to be an efficient few-shot learner.

The main work of this paper has the following three points:

- We proposed a zero-shot event detection model based on ordered contrastive learning. By constructing multiple contrastive samples with ordered similarities, the encoder could learn a better representation of unseen types.

- We creatively introduce the prompt-based prediction into the zero-shot event detection problem for trigger words identification, which removed the dependency on predefined event structures and heuristic rules.

- Experiments on two English datasets demonstrate that the both supervised and zero-shot event detection performance are improved via ordered contrastive learning and prompt-based prediction.

2 Related Work

2.1 Zero-shot event detection

The transfer learning-based zero-shot methods mainly rely on a predefined event structure as heuristic rules. In other words, models must know the unseen event name (e.g Attack) and the elements (e.g Attacker) consist of unseen events. Huang et al. (2018) and Zhang et al. (2021b) cluster unseen events by label semantic similarity with the help of semantic structures analyzing tools such as Abstract Meaning Representation (AMR) or Semantic Role Labeling (SRL). Additionally, Lyu et al. (2021) need to define QA queries for unseen event types manually. Although Huang and Ji (2020) proposed SS-VQ-VAE to discover new event types without human assistance, external part-of-speech tagging tool is needed to find candidate trigger words. Meanwhile, the semi-supervised loss function could only roughly separate all unseen event samples from seen events. And the variational autoencoder focuses on the feature learning of single instance. So it is still challenging to cluster unseen events into multiple new types.

2.2 Contrastive learning

Contrastive learning aims to learn high-quality feature representations through self-supervision. The core of contrastive learning is constructing positive and negative sample pairs. For labeled data, the construction is relatively simple, where random sampling by supervise label works in most cases. For unsupervised learning with unlabeled data, it needs more strategy to construct sample pairs. Wang et al. (2021); Logeswaran and Lee (2018) treat the target sentence's context as a positive sample. Wang et al. (2020) proposed various sample editing methods based on word masking and shuffle. Gao et al. (2021b) introduce a dropout mask for constructing contrastive samples, which doesn't need any textual edit. These approaches focus on instance-level feature contrastive and only divide samples into positive and negative. Zhang et al. (2021a) try to overcome this weakness by optimizing a top-down clustering loss. Considering that class-level features learning is as essential as instance-level features, it's necessary to improve the traditional contrastive learning framework for the zero-shot event detection task.

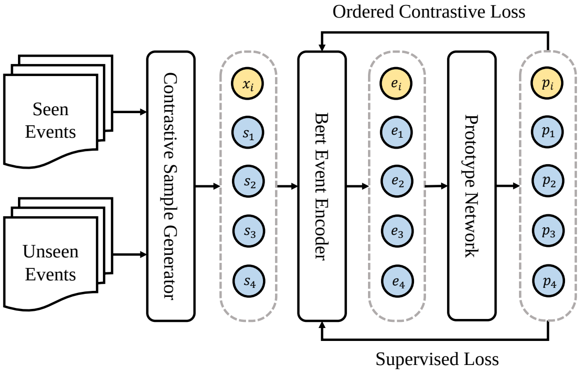

Figure 3: The architecture overview of ZEOP.

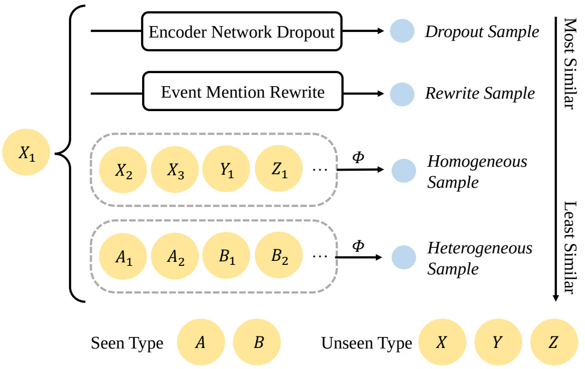

Figure 4: How contrastive samples are constructed for unseen types. The Φ denotes random sample operation.

2.3 Prompt-based prediction

Prompt-based prediction (Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021a) treats the NLP downstream task as a masked language modeling problem. The language model first generates a label word to a given prompt defined by a task-specific template. Then the label word is mapped to downstream task output space. In this way, knowledge can be extracted from pretrained language models at low cost, which makes full use of pre-training corpus. This is an ideal approach for event trigger word identification in the zero-shot event detection task scenario, because it doesn't relay on any heuristic rules or external tools as multiple approaches mention in Section 2.1,

3 Methodology

The architecture overview of **Z**ero-Shot **E**vent Detection Based on **O**rdered Contrastive Learning and **P**rompt-Based Prediction (ZEOP) is shown in Figure 3. Given a set of seen events S and unseen events U, all samples are first input into the contrastive sample generator as the original sample x_i, where a list of multiple contrastive samples is constructed as $\{s_1, \ldots, s_4\}$. Subsequently, the event encoder encodes event mentions as embedding vector e_i, and the prototypical network predicts probability distribution over event types as p_i. Ordered contrastive loss is calculated for all samples, and supervised loss is calculated only for samples of seen events. Model parameters in the event encoder and prototypical network could be updated by gradient backward.

3.1 Contrastive sample generator

In this paper, we aim to resolve the zero-shot event detection task. On the one hand, seen events samples could offer supervised labels, which are ideal class-level contrastive samples. On the other hand, there are also some unseen events samples without supervised labels, which could only be used for instance-level contrastive samples. Therefore, we construct four constructive samples, including both class-level and instance-level, which are shown in Figure 4. The similarities between these contrastive and original samples differ from strong to weak.

3.1.1 Dropout sample

Dropout mask is proposed by (Gao et al., 2021b), which passes the same sentence to the pre-trained encoder twice. Because the network nodes of the encoder are randomly dropped when training, a different event embedding will be obtained for the second time. The dropout sample should be considered the most similar contrastive sample with the same input sentences.

3.1.2 Rewrite sample

The rewrite sample is textually edited from the original event mention. In order to ensure the consistency of semantics, we choose the back translation (Fadaee and Monz, 2018) for event mention rewriting. Rewrite samples should be considered the secondary similar contrastive sample because they keep the original event trigger and elements.

3.1.3 Homogeneous sample

Homogeneous samples are events with the homogeneous event type. For seen events, the homogeneous samples are random sampled from events with same label $\{s_i \in S | s_i.type = x_i.type\}$. For unseen types, the label of the original event is invisible for the model, so the homogeneous samples are randomly sampled from all unseen types U.

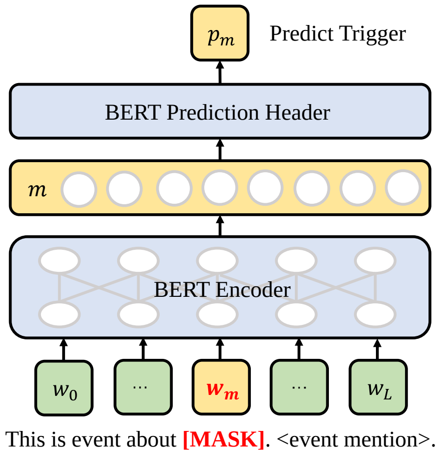

Figure 5: Prompt-based trigger word prediction. The prompt template is populated with the event mention and encoded by a pre-trained BERT. The contextual vector corresponding to the [MASK] tag is used for trigger prediction.

3.1.4 Heterogeneous sample

Heterogeneous samples are events with heterogeneous types. For seen events, heterogeneous types mean seen events with a different label $\{s_i \in S | s_i.type \neq x_i.type\}$. For unseen events, the heterogeneous type means all seen types S due to the label of the original event is invisible for the model.

It should be noted that a homogeneous sample may have a different label with the original sample. But it could be guaranteed that heterogeneous samples always have different label. Therefore the heterogeneous samples are regarded as the least similar contrastive samples.

3.2 Event encoder

Following existing zero-shot event detection approaches (Huang and Ji, 2020; Zhang et al., 2021b), the best embedding feature for an event should be the contextual vector of trigger words. The problem is how to identify the trigger words under the zero-shot setup. Inspired by (Brown et al., 2020; Schick and Schütze, 2021; Gao et al., 2021a), we utilize the prompt-based prediction for trigger word identification, which doesn't rely on any heuristic rules or external semantic analyzing tools.

3.2.1 Trigger word prediction

As shown in Figure 5, we use "This is event about [MASK]. <event mention>" as prompt template, where the [MASK] is the trigger word that pre-trained BERT (Devlin et al., 2019) language model should predict, and the <event mention> is the text describing an event. In specific, given a BERT input sequence $t = \{w_0, w_1, w_2, \dots, w_L\}$, where w_i is the ith token of the template sentence, we obtain a word distribution $p_m(w_m = w_i|t)$ over all the words in the event mention.

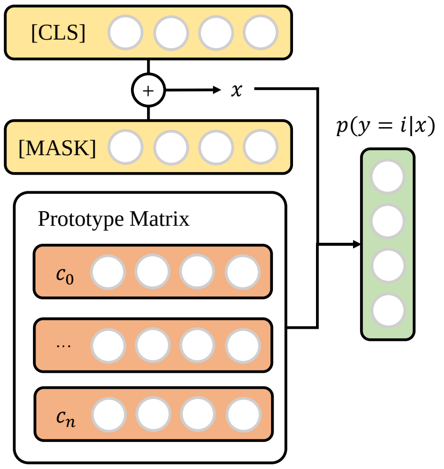

Figure 6: Event type prediction by prototype network. The query vector is the result of element-wise adding the contextual vector of [CLS] and [MASK] token.

3.2.2 Event type prediction

For event type prediction, we introduce prototype network (Snell et al., 2017). It defines a prototype matrix $C \in \mathbb{R}^{n \times h}$, where each row represents the prototype of one embedded event type c_i and h is the embedding dim of BERT. The number of embedding types is $n = k + l$, where k is the number of seen event types and l is the number of unseen event types. The value of the prototype matrix is randomly initialized and keeps updating while training the model. As shown in Figure 6, give a query vector x, the distribution over event types is calculated by the prototypical network as follow:

$$p(y = i \,|\, x) = \frac{\exp(-d(f(x), c_i))}{\sum_{i'} \exp(-d(f(x), c_{i'}))} \tag{1}$$

where $d(x, c_i)$ is the Euclidean distance between embedding vector x and c_i.

One event type may correspond to multiple trigger words in the event detection task. If only the predicted trigger words were used as the query point, samples with the same event type would be mapped to hidden space with large distances. So we add the contextual vector of [CLS] and [MASK] token as query vector x to balance event type and trigger words features.

3.3 Ordered contrastive loss

The design goal of the contrastive loss function is to narrow the distance between similar samples and push away different samples. To better represent unseen events under the zero-shot setting, the proposed ordered contrastive learning constructs two represent contrastive samples (Dropout Sample and Rewrite Sample), an inner-cluster contrastive sample (Homogeneous Sample), and an inter-cluster contrastive sample (Heterogeneous Sample). Given four contrastive samples with different similarities to the original sample, we designed a novel ordered contrastive loss function by expanding the contrast loss function in (Hadsell et al., 2006). Our model could learn the partial order relationship in similarity: Dropout Sample > Rewrite Sample > Homogeneous Sample > Heterogeneous Sample. Thus model will distinguish between different unseen type events more significant.

Let p_0, p_1, p_2, p_3, and p_4 be the event type probability distribution of the original, dropout, rewrite, homogeneous, and heterogeneous samples, respectively. The distance between contrastive samples and the original sample are calculated as d_1, d_2, d_3, and d_4:

$$d_i = W_p(p_i, p_0), i \in \{1,2,3,4\} \tag{2}$$

Considering the vectors to be compared are probability distribution, we utilize Wasserstein distance (Kolouri et al., 2019) as the distance function W_p. Compared to Euclidean distance and cosine distance commonly used in contrastive learning, Wasserstein distance could better measure the difference between probability distributions.

$$W_p(\mu, \nu) = \left(\inf_{\gamma \in \Gamma(\mu,\nu)} \int_{X \times Y} d^p(x,y) d\gamma(x,y) \right)^{\frac{1}{p}} \tag{3}$$

The similarities between contrastive samples and the original sample decrease. Four distances may form an increasing list. For seen events, this list is strictly increasing. For unseen events, this list is not strictly increasing because the homogeneous sample may carry a different label. Therefore, the ordered contrastive loss for sample x is calculated as:

$$\begin{aligned} L_c = d_1 + L_m(d_2, d_1) + L_m(d_3, d_2) \\ + \begin{cases} L_m(d_4, d_3) & x \in S \\ L_m(d_4, d_2) & x \in U \end{cases} \end{aligned} \tag{4}$$

Table 1: Statistics of datasets. The $|V|$ and $|D|$ are the number of samples and event types. The last two rows are the mean and standard deviation of samples by type.

Dataset	ACE-2005		FewShotED	
	$\lvert V\rvert$	$\lvert D\rvert$	$\lvert V\rvert$	$\lvert D\rvert$
Seen	2316	17	40893	50
Unseen	1489	16	33439	50
Total	3805	33	74332	100
Mean	115.30		743.32	
Stdev	206.32		2828.47	

where $L_m(d_x, d_y)$ is the margin loss:

$$L_m(d_x, d_y) = max(0, margin - (d_x - d_y)) \tag{5}$$

3.4 Supervised loss

Since the first k rows of the prototype matrix correspond to the seen event types, we could also apply supervised learning for seen events in addition to the contrastive learning. With the label and trigger word visible to the model, the negative log loss function is calculated as:

$$L_s = \begin{cases} -\hat{y}_x log(y_x) - \hat{z}_x log(z_x) & x \in S \\ 0 & x \in U \end{cases} \tag{6}$$

where $\hat{y}_x$ and $\hat{z}_x$ are the ground truth label of event type and trigger words, respectively.

The complete loss of ZEOP is the sum of ordered contrastive loss and supervised loss

$$Loss = L_c + L_s \tag{7}$$

4 Experiments

4.1 Implementation

We implement our model in PyTorch (Paszke et al., 2019) with Transformer Library (Wolf et al., 2020) and choose *bert-base-uncased* as the pre-trained language model. For back translation, we use Argos Translate[1] and set Chinese as the intermediate language. For model training, we use the AdamW(Loshchilov and Hutter, 2019) optimizer with batch size of 32, and the learning rate is grid searched in $[1e-7, 1e-4]$ for parameters of BERT, $[1e-4, 1e-2]$ for non-BERT parameters. The margin in contrastive loss is set to 1, and the unseen event types count l is set to the actual value of the dataset. With above settings, the total number of parameters in ZEOP is 109.53 million, of

[1] https://www.argosopentech.com

Table 2: Comparison with different baseline. The results are averaged across 3 runs with random seed 2020, 2021, and 2022.

Model	ACE-2005				FewShotED			
	F1-Seen	F1-Unseen	NMI	FM	F1-Seen	F1-Unseen	NMI	FM
SCCL	0.5999	0.3190	0.3259	0.2403	0.8717	0.3640	0.2647	0.3462
SS-VQ-VAE	0.6988	0.3509	0.2515	0.4269	0.9208	0.4364	0.1722	0.5762
BERT-OCL	0.6040	0.3751	**0.4532**	0.2551	0.9017	0.2160	0.4157	0.1894
ZEO	0.7566	0.4230	0.3771	0.4253	**0.9361**	0.5456	0.4792	0.6410
ZEOP	**0.7771**	**0.4591**	0.3797	**0.4913**	0.9306	**0.5814**	**0.4831**	**0.7139**

which 109.51 million (99.98%) belong to the pretrained BERT. All the experiments are performed on a Linux server with four RTX 3090 GPUs. The code of all experiments is available at GitHub[2].

4.2 Dataset

We evaluate the proposed model on two datasets in English. Ace-2005[3] is a widely used (Huang et al., 2018; Huang and Ji, 2020; Zhang et al., 2021b; Lyu et al., 2021) dataset for the event detection task. FewShotED (Deng et al., 2020) is a dataset proposed for the few-shot event detection task. To balance the sample count between seen and unseen types, we first sort the event types by the decreasing order of the sample count. Then take event types at the odd position as seen types and the even position as the unseen type. For example, given a list of event type list t_1, t_2, t_3, t_4 sorted by sample count, the t_1 and t_3 will be mark as seen event types, the t_2 and t_4 will be mark as unseen event types. The Statistics of the processed dataset are shown in table 1. We may assume that each sample only belong to one event type. In comparison, the number of samples and types in the ACE-2005 dataset is smaller than FewShotED. The problem of sampling bias is more evident on FewShotED. These datasets will be randomly divided into training set, validation set, and test set at a ratio of 8:1:1.

4.3 Evaluation

We set up two tasks for evaluation: supervised event detection for seen events and zero-shot event detection for unseen events. The f1 score will be used as the common metric for two tasks. For seen events, the predicted labels are directly output by the model. For unseen events, the predicted labels are mapped from model outputs by the Hungarian Algorithm. Additionally, following (Huang and Ji, 2020; Zhang et al., 2021a), Normalized Mutual Info (NMI) and Fowlkes Mallows (FM) will be used for unseen events detection to evaluate the clustering performance.

4.4 Baseline

Considering that the zero-shot event detection task focused in this study has no available pre-defined event types, many approaches mentioned in Section 2.1 are not applicable because they require event type names or QA queries as model input. So we use the following approaches as the experimental baselines.

- **SCCL** (Zhang et al., 2021a): A state-of-the-art model designed for unsupervised text clustering could detect new event types from unseen event mentions. We use the contextual vector of same candidate trigger words as SS-VQ-VAE instead of [CLS] token as event mention representation to fit event detection task.

- **SS-VQ-VAE** (Huang and Ji, 2020): A semi-supervised zero-shot event Detection model uses the variational autoencoder as regularizer. It considers all noun and verb concepts that can be mapped to OntoNotes senses as candidate trigger words.

- **BERT-OCL**: We fine-tune a BERT with the ordered contrastive framework proposed in this paper, where the distance between the contrastive sample and the original sample is calculated by Euclidean distance. Once event encoding is obtained, KNN algorithm is applied to seen events detection as the classifier and K-means algorithm is applied to unseen event detection as the cluster.

- **ZEO**: The version of ZEOP don't identify event trigger word by prompt-based prediction, and use the same heuristic rules as SS-VQ-VAE to identify candidate trigger words.

[2] https://github.com/KindRoach/NAACL-ZEOP

[3] https://catalog.ldc.upenn.edu/LDC2006T06

Table 3: Ablation study for different contrastive samples.

Model	ACE-2005				FewShotED			
	F1-Seen	F1-Unseen	NMI	FM	F1-Seen	F1-Unseen	NMI	FM
ZEOP-woCL	**0.8365**	0.4082	0.3477	0.4523	**0.9658**	0.5078	0.4803	0.5396
+Dropout	0.8128	0.4219	0.3789	**0.5545**	0.9650	0.5467	0.4637	**0.7263**
+Rewrite	0.8279	0.3732	0.3382	0.4994	0.9591	0.5402	0.4743	0.7246
+Homogeneous	0.7433	0.4238	0.3569	0.5463	0.9378	0.5625	0.4453	0.6741
ZEOP	0.7771	**0.4591**	**0.3797**	0.4913	0.9306	**0.5814**	**0.4831**	0.7139

Table 4: Ablation study for different contrastive distance metrics.

Model	ACE-2005				FewShotED			
	F1-Seen	F1-Unseen	NMI	FM	F1-Seen	F1-Unseen	NMI	FM
ZEOP-Eu	**0.8170**	0.3533	0.3465	0.3152	**0.9648**	0.4505	0.4656	0.4770
ZEOP-Kl	0.7910	0.3543	**0.3960**	0.2515	0.9245	0.3236	0.4465	0.3188
ZEOP	0.7771	**0.4591**	0.3797	**0.4913**	0.9306	**0.5814**	**0.4831**	**0.7139**

4.5 Overall performance

The overall performance of ZEOP and all baseline approaches are shown in Table 2. Our proposed model achieved the best overall performance on two datasets for both seen and unseen event detection tasks, except the BERT-OCL takes the lead on Normalized Mutual Info on ACE-2005, which may be due to the small number of samples in the dataset. The prototypical network used by ZEOP needs more samples to train its prototype matrix of classes, while Bert-OCL could directly use the embedding result of pre-trained bert. Besides, we could observe that: 1) Comparing the evaluation of ZEO and ZEOP, although ZEO shows a slight performance advantage on F1 score on FewShotED for seen event detection task, other metrics proved that prompt-based prediction gives model better performance than heuristic rules as trigger word identifier. 2) The SCCL and BERT-OCL show worse performance than SS-VQ-VAE on seen event detection task, which demonstrates that supervised learning is still necessary and could not be replaced by contrastive learning.

4.6 Ablation study

To explore the effect of four contrastive samples, we conduct an ablation experiment by introducing them one by one to the ZEOP-woCL model, which takes no contrastive samples. The comparison of their performance is shown in Table 3. As all contrastive samples are added to the model, the performance improves on the unseen event detection task, but there is a performance decrease on the seen event detection task. This demonstrates that contrastive learning does help the model to learn a better representation of unseen types, but its training objectives will conflict with the goal of supervised learning.

What's more, we also validate the effect of Wasserstein distance as the distance metric in the ordered contrastive loss. Table 4 indicates that the ZEOP model using Wasserstein distance performs better on unseen event detection task than ZEOP-Eu using Euclidean distance and ZEOP-Kl using Kullback-Leibler divergence. However, Euclidean distance shows better results on seen event detection task.

4.7 Qualitative analysis

For qualitative analysis, we visualize all unseen types samples in the dataset by t-SNE [4]. As Figure 7 and Figure 8 show, the model with ordered contrastive learning (ZEOP and BERT-OCL) could learn a better representation of unseen types than other baselines. The larger the number of samples a type contains, the better cluster will be achieved. Overall, the classification of unknown event types is significantly more complex than known types, and only a small number of types with large sample sizes are correctly classified. Many clusters could not be mapped one-to-one to the actual event type with the absence of supervision learning. The model may use event elements such as subject and location as clustering features rather than the event type itself.

4.8 Hyperparameter sensitivity analysis

The number of clusters is an essential hyperparameter in the clustering problem. In order to explore

[4] https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

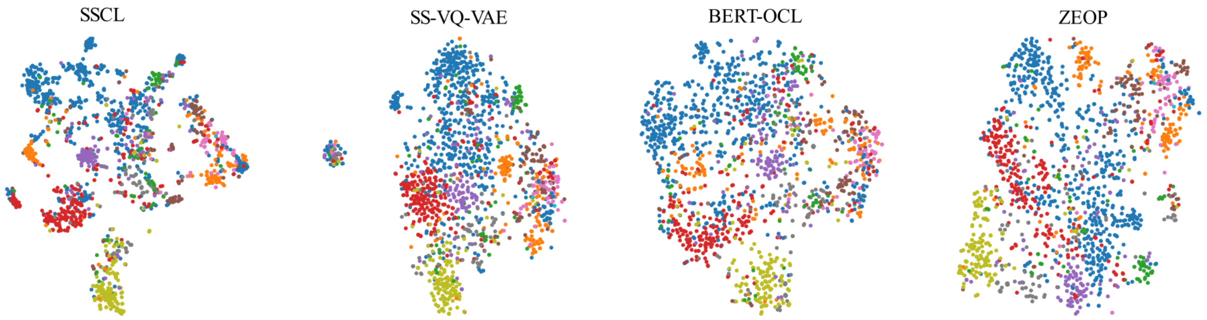

Figure 7: Visualization of unseen types on ACE-2005. Each color indicates a ground truth type.

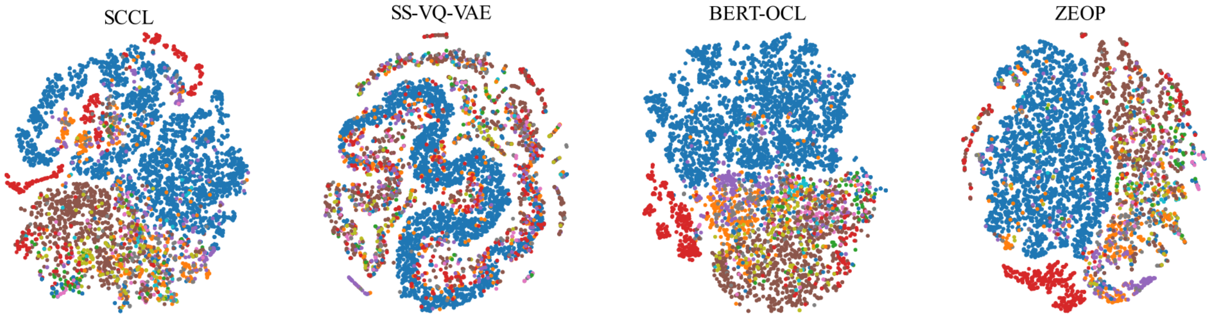

Figure 8: Visualization of unseen types on FewShorED. Each color indicates a ground truth type.

the impact of parameter unseen types l on the performance of ZEOP, we conduct a hyperparameter experiment by setting l to the 1, 2, 3, 4, and 5 times of the actual number of unseen types. As shown in Figure 9, ZEOP suffers a slight performance drop when unseen type numbers increase. The model is more likely to classify the same event type into different clusters. In practice, properly estimating the approximate number range of unknown event types will help the model to achieve a better performance.

5 Conclusion

In order to solve the problem of zero-shot event detection, this paper proposes an end-to-end model named Zero-Shot Event Detection Based on Ordered Contrastive Learning and Prompt-Based Prediction. By creatively introducing multiple contrastive samples with different similarities, the contrastive loss is extended from pairwise comparison to list-wise comparison. Therefore, the model could learn a better representation across instance-level and class-level. Meanwhile, the prompt-based prediction is utilized to identify event trigger words without relying on heuristic rules. Experiments demonstrate that our method can significantly improve the accuracy of identifying unseen event types while keeping the ability to classify seen event types. Future research should consider the po-

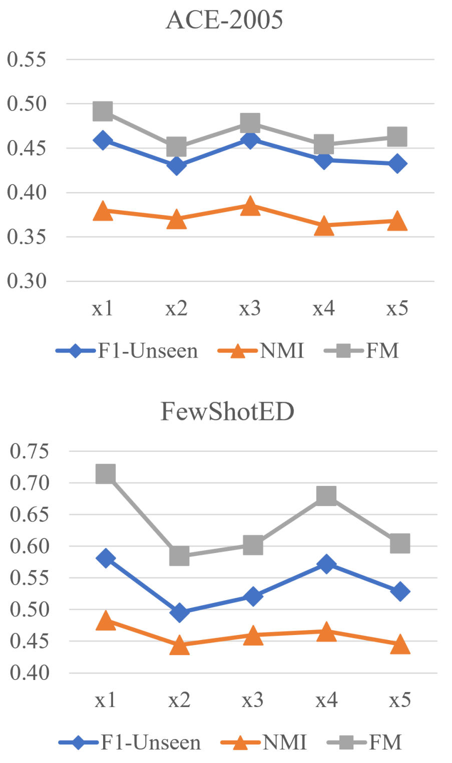

Figure 9: Effects of unseen types number.

tential effects of the initialization of the prototype matrix in the prototypical network more carefully. A better initial value may reduce the need for large training samples and speed up model training.

6 Acknowledgments

This work was supported by NSFC grants (No. 61972155), the Science and Technology Commission of Shanghai Municipality (20DZ1100300) and the Open Project Fund from Shenzhen Institute of Artificial Intelligence and Robotics for Society, under Grant No. AC01202005020, Shanghai Trusted Industry Internet Software Collaborative Innovation Center.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, and etc. 2020. Language models are few-shot learners. In *NeurIPS*.

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 *WSDM*, 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 *NAACL*, pages 4171–4186.

Marzieh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. In *EMNLP*, pages 436–446.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021a. Making pre-trained language models better few-shot learners. In *ACL*, pages 3816–3830.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021b. Simcse: Simple contrastive learning of sentence embeddings.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *CVPR*, pages 1735–1742.

Lifu Huang and Heng Ji. 2020. Semi-supervised new event type induction and event detection. In *EMNLP*, pages 718–724.

Lifu Huang, Heng Ji, Kyunghyun Cho, Ido Dagan, Sebastian Riedel, and Clare R. Voss. 2018. Zero-shot transfer learning for event extraction. In *ACL*, pages 2160–2170.

Soheil Kolouri, Kimia Nadjahi, Umut Simsekli, Roland Badeau, and Gustavo K. Rohde. 2019. Generalized sliced wasserstein distances. In *NeurIPS*, pages 261–272.

Ying Lin, Heng Ji, Fei Huang, and Lingfei Wu. 2020. A joint neural model for information extraction with global features. In *ACL*, pages 7999–8009.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. In *ICLR*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *ICLR*.

Qing Lyu, Hongming Zhang, Elior Sulem, and Dan Roth. 2021. Zero-shot event extraction via transfer learning: Challenges and insights. In *ACL*, pages 322–332.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*, pages 5900–5907.

Adam Paszke, Sam Gross, Francisco Massa, and etc. 2019. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *NAACL*, pages 2339–2352.

Jake Snell, Kevin Swersky, and Richard S. Zemel. 2017. Prototypical networks for few-shot learning. In *NeurIPS*, pages 4077–4087.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *EMNLP*, pages 5783–5788.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Junchi Yan, Peng Gao, and Guotong Xie. 2020. Pre-training entity relation encoder with intra-span and inter-span information. In *EMNLP*, pages 1692–1705.

Ziqi Wang, Xiaozhi Wang, Xu Han, Yankai Lin, Lei Hou, Zhiyuan Liu, Peng Li, Juanzi Li, and Jie Zhou. 2021. CLEVE: contrastive pre-training for event extraction. In *ACL*, pages 6283–6297.

Thomas Wolf, Lysandre Debut, Victor Sanh, and etc. 2020. Transformers: State-of-the-art natural language processing. In *EMNLP*, pages 38–45. Association for Computational Linguistics.

Congle Zhang, Stephen Soderland, and Daniel S. Weld. 2015. Exploiting parallel news streams for unsupervised event extraction. *Trans. Assoc. Comput. Linguistics*, 3:117–129.

Dejiao Zhang, Feng Nan, Xiaokai Wei, Shang-Wen Li, Henghui Zhu, Kathleen R. McKeown, Ramesh Nallapati, Andrew O. Arnold, and Bing Xiang. 2021a. Supporting clustering with contrastive learning. In *NAACL*, pages 5419–5430.

Hongming Zhang, Haoyu Wang, and Dan Roth. 2021b. Zero-shot label-aware event trigger and argument classification. In *ACL*, pages 1331–1340.