mathematics

MDPI

Article

Learning Event Representations for Zero-Shot Detection via Dual-Contrastive Prompting

Jiaxu Li, Bin Ge *, Hao Xu, Peixin Huang and Hongbin Huang

Laboratory for Big Data and Decision, National University of Defense Technology, Changsha 410073, China; lijiaxu18@nudt.edu.cn (J.L.); xuhao@nudt.edu.cn (H.X.); huangpeixin15@nudt.edu.cn (P.H.); hbhuang@nudt.edu.cn (H.H.)

* Correspondence: gebin@nudt.edu.cn

Abstract: Zero-shot event detection aims to involve the automatic discovery and classification of new events within unstructured text. Current zero-shot event detection methods have not considered addressing the problem more effectively from the perspective of improving event representations. In this paper, we propose dual-contrastive prompting (COPE) model for learning event representations to address zero-shot event detection, which leverages prompts to assist in generating event embeddings using a pretrained language model, and employs a contrastive fusion approach to capture complex interaction information between trigger representations and sentence embeddings to obtain enhanced event representations. Firstly, we introduce a sample generator to create ordered contrastive sample sequences with varying degrees of similarity for each event instance, aiding the model in better distinguishing different types of events. Secondly, we design two distinct prompts to obtain trigger representations and event sentence embeddings separately. Thirdly, we employ a contrastive fusion module, where trigger representations and event sentence embeddings interactively fuse in vector space to generate the final event representations. Experiments show that our model is more effective than the most advanced methods.

Keywords: event detection; event representations; dual-contrastive learning; contrastive fusion

MSC: 68T30; 68T50

Citation: Li, J.; Ge, B.; Xu, H.; Huang, P.; Huang, H. Learning Event Representations for Zero-Shot Detection via Dual-Contrastive Prompting. *Mathematics* **2024**, *12*, 1372. https://doi.org/10.3390/math12091372

Academic Editor: Pedro A. Castillo Valdivieso

Received: 1 April 2024
Revised: 26 April 2024
Accepted: 26 April 2024
Published: 30 April 2024

1. Introduction

Event detection is used to identify event triggers and classify events from natural language texts. Conventional *supervised* event detection methods [1–3] rely on a large number of labeled instances, in order to recognize an event and put it into a *seen* event type. To enable the model to cope with unseen event types, there is a recent trend to investigate event detection in zero-shot scenarios, which discovers and classifies new events from texts without annotations. Specifically in this configuration, events are divided into *seen* and *unseen* types [4–6], in which "seen" means that the model can see the label information of these event types during the training process, while "unseen" implies that the model has not seen the label information of these types during training.

Figure 1 illustrates the task of event detection in zero-shot scenarios (to be formally defined in Section 3). For a seen event, from the sentence *S1* "*The Daily Planet raised 2.2 million US dollars in its initial public offering with one of the new 600 shareholders acquiring 1.0 million dollars worth of shares*", which is also referred to as an event sample, it is used to identify the event trigger word "*acquiring*", and classify the event into type `Transaction.Transfer-Ownership`. Further, for a new unseen type event, from the sentence *S2* "*Cash-strapped Vivendi wants to sell Universal Studios, its Universal theme parks and television production company*", it is also used to detect its trigger word "*meeting*", but classify it into a new event type *N* (when there is another event sample of the same type as *S2*, it will also be put as an event of type *N*).

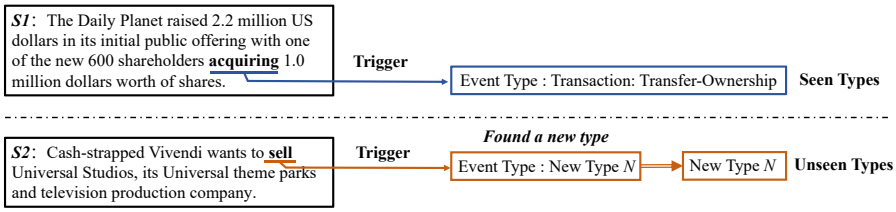


Figure 1. Illustration of zero-shot event detection task. Case for seen (resp. unseen) event types above (resp. below) the dashed center line.

While recent years have witnessed progress in this task, conventional approaches rely on pre-defined event types as heuristic rules [7,8] or external knowledge bases [5,9], which may not be always available in practice, and hence limit their applications. Lately, taking no such information for granted, methods in pursuit of better event representations have been sought. High-quality representation is shown to be an effective means that benefits various event-centric tasks [10–14].

In particular, Zhang et al. [15] proposed, among the first, a novel approach, =namely ZEOP, to tackle zero-shot ED by leveraging prompt tuning and contrastive learning. In ZEOP, prompt tuning is incorporated to generate the `[MASK]` token embedding for determining trigger words, and this task-specific embedding, combined with the task-agnostic embedding (i.e., the `[CLS]` token embedding) from BERT, is then used for event classification. In short, two types of event representations—one task-specific (for event trigger recognition) and the other task-agnostic (for general event-centric tasks)—are fused together for the prediction.

Albeit viable, ZEOP falls short in two aspects: (1) the straightforward combination of embeddings may not well fit the task of zero-shot ED, exhibiting suboptimal performance, since the `[MASK]` token embedding is rather task-specific for trigger recognition, which may not be beneficial to zero-shot ED, while the `[CLS]` token embedding is too general; and (2) a primitive sentence embedding directly from BERT is employed for feature fusion, which can be less effective for zero-shot ED, due to the unnecessary influence from static token embedding biases and ineffective layers [16]. This motivates us to seek a better fusion mechanism and more effective representations for zero-shot ED.

In this research, we propose a dual-contrastive learning model, namely COPE, which works with two levels of contrastive learning. Firstly, it adopts ordered contrastive learning [15], where augmented samples of varying *similarities* are exploited to produce a *sample*-level contrastive loss. On top of that, it further carries out a contrastive fusion learning, where event embeddings of varying *utilities* via different representation mechanisms are constructed to produce an *instance*-level contrastive loss. In our design, the sample-level contrast enables the model to discriminate events, and the instance-level contrast makes the model learn to strike a good balance in combining task-specific and task-agnostic embeddings.

Specifically in the instance-level contrastive learning, we put forward two new prompts (i.e., TR-prompt and ER-prompt) to obtain two kinds of representations, respectively, from the perspectives of event triggers and background sentences. To combine them into the final event representation for zero-shot ED, we devise a contrastive fusion strategy, which forces the representation concerning event trigger (by TR-prompt) close to the representation of background sentence concerning event type (by ER-prompt) while far away from that of the sentence (from direct BERT encoding). In this way, the complex interaction between trigger information and the overall event is to be captured in the high-dimensional vector space. Afterwards, we adopts a prototype network for event type classification.

In summary, the main contribution of this article is at least three-fold:

- We propose a new method for zero-shot ED, namely COPE, which leverages a dual-contrastive learning framework, i.e., sample-level and instance-level, for learning better event representations pertinent to the task;

- We conceive a contrastive fusion strategy to capture the complex interaction information within events from two perspectives—event triggers and background sentences—such that a balance between task-specific and task-agnostic features is achieved in embedding fusion;
- We validate the performance of our model on two benchmark datasets, and the experiment results indicate that COPE offers superior performance in both seen and unseen event detection, in comparison with state-of-the-art models.

2. Related Work

In this section, we first introduce the related work on zero-shot ED. Then, based on the methods employed, we provide an overview of the related work on event representation learning and contrastive learning.

2.1. Zero-Shot Event Detection

The core challenge of zero-shot learning is the significant difference in representations between seen and unseen types in the feature space. The feature partitions learned by traditional deep learning models on seen types are difficult to directly apply to unseen types. In recent years, various zero-shot event detection algorithms have been proposed, and based on their core design principles, they can be classified into two categories: methods based on transfer learning and semi-supervised methods.

Huang et al. [7] and Zhang et al. [8] used transfer learning for zero-shot event detection, but it relies on artificially defined event structures as heuristic rules. Similarly, Lyu et al. [9] proposed a method needs to manually define TE or QA for unseen event types to accomplish knowledge transfer between different event types. Huang and Ji [5] proposed SS-VQ-VAE to discover new event types without human assistance. On the basis of SS-VQ-VAE, Zhang et al. [15] added ordered contrast learning and trigger prediction prompt, which can complete event detection tasks without relying on external resources. These methods encode events using the original pretrained language model during the acquisition of event representations, striving to differentiate between seen and unseen types to the best of their ability.

2.2. Event Representation Learning

Event representation learning aims to automatically learn semantic feature representations of events from large-scale data and supports the model's further use in data training and prediction. Early event representation learning primarily employed a neural tensor network [17] to acquire event representations by learning the semantic composition of events [18,19]. However, these methods introduce component induction bias and cannot be extended to new events [20].

Many studies achieved strong performance in representation learning [21,22] with pretrained language models in both supervised and unsupervised settings. Recent studies have replaced static word vector compositions with pretrained language models [23,24], like BERT, to obtain flexible event representations, resulting in improved performance in downstream tasks. Although BERT achieved success in representation learning, original BERT shows unsatisfactory performance [25,26], primarily due to the occurrence of the "cone effect" in sentence embeddings, which is influenced by token frequency. The latest research uses the PromptBERT method [16], and the output of sentence embedding is not the label predicted by the MLM classification head, but the vector representing the sentence. This method effectively uses the original BERT layer by using large-scale knowledge and avoid embedding bias.

2.3. Contrastive Learning

Contrastive learning is a self-supervised learning method. The core idea is to help the model learn high-quality feature representations by constructing contrastive samples. For event detection tasks, we aim to aggregate events of the same type together and separate

events of different types as much as possible through contrastive learning. Logeswaran and Lee [27] regard the context of the target sentence as a positive sample. The dropout mask is proposed by Gao et al. [21], which uses the characteristics of the dropout layer to obtain the most similar contrast sample to the original sample. These methods only divide the samples into positive samples and negative samples. Zhang et al. [15] introduced homogeneous sample and heterogeneous sample, constructed ordered contrastive samples, and better distinguished different types by learning the partial order relationship of different contrastive samples.

The above-mentioned methods have achieved success in instance-level contrastive sample construction. However, for zero-shot event detection, instance-level contrastive learning is insufficient to assist the model in better distinguishing features of different events due to the lack of supervised signals for corresponding event types [20]. Additionally, contrastive samples may introduce some noise. Therefore, this paper not only considers instance-level contrastive learning but also incorporates representation-level contrastive learning specifically for event representations. This involves utilizing the features of event representations to differentiate between different types of events.

3. Problem Definition

Let $\varepsilon_s = \{e_1, e_2, \ldots, e_k\}$ denote a set of seen event types and $\varepsilon_u = \{e_1, e_2, \ldots, e_l\}$ denote a set of unseen event types; the set of all event types is $\varepsilon = \varepsilon_s + \varepsilon_u$. We define the seen event set $D_s = \{(x_i, y_i), y_i \in \varepsilon_s\}$ and the unseen event set $D_u = \{(x_i)\}$. The goal of the zero-shot event detection task is, given an input event x, to predict its probability distribution $p_i = p(y = e_i, e_i \in \varepsilon | x)$ on the event type. If the input is a seen event, then $p_i = p(y = e_i, e_i \in \varepsilon_s | x)$; this is a conventional supervised event detection setting. If the input is an unseen event, then $p_i = p(y = e_i, e_i \in \varepsilon_u | x)$. Particularly, for unseen events, as the ε_u is not visible to the model, the prediction of event types does not have a one-to-one correspondence with their true types. At this point, event detection resembles more of a clustering task.

4. Methodology

We propose a prompt-based dual-contrastive representation learning method for zero-shot event detection, as illustrated in Figure 2. The method is mainly divided into three modules: contrastive sample generation, event representation learning, and event type prediction. In event representation learning, we utilize a contrastive fusion approach to achieve better event representations. We will detail the working mechanisms of each module in the following sections.

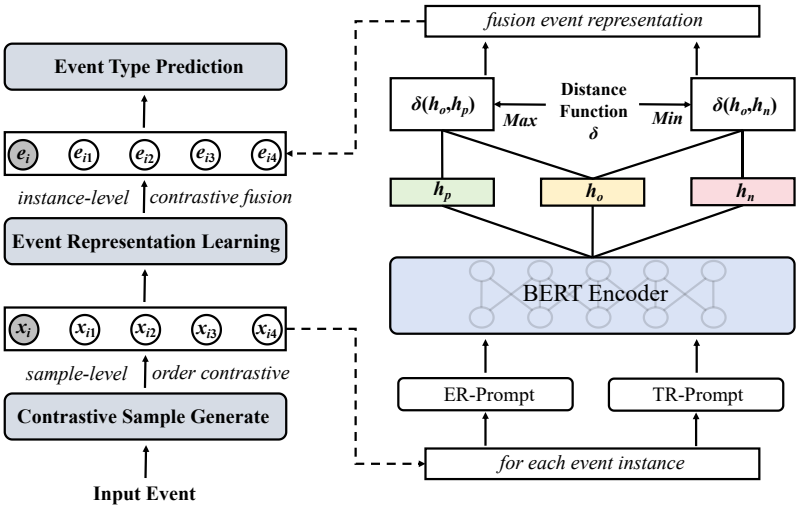


Figure 2. Overview of COPE. The left part represents the overall flow of the model. The direction of the dashed line indicates the actual process of samples passing through representation learning to obtain event representations. The gray-marked section represents the original event input and the encoded results of the original events. The prompts here are simplified and shortened due to the space limit.

4.1. Contrastive Sample Generator

Following the previous work [15], we use the same method to construct multiple contrastive samples for the original samples. For each input sample x_i, the sample generator generates four different samples.

Dropout sample. This is obtained by the dropout layer in the encoder network [21]. Since the encoder will randomly drop out the network nodes during training, the same event will obtain different embeddings in the second entry into the encoder.

Rewrite sample. Considering that the event contains multiple feature information such as trigger words and arguments, when editing the original text, it is necessary to ensure that the semantics of the original event are not changed, so the sample is obtained by back translation.

Homogeneous sample. It aims to obtain samples of the same type as the original samples. For $x_i \in D_s$, contrastive sample x_i' can be randomly sampled from D_s, where $y_i' = y_i$. For $x_i \in D_u$, contrastive sample x_i' can be randomly sampled from D_u, since for all unseen types, the labels of the original events are invisible to the model, so all unseen events can be considered as "the same type".

Heterogeneous sample. Contrary to the homogeneous sample, the objective is to obtain samples with the lowest similarity to the original samples. For $x_i \in D_s$, contrastive sample x_i' can be randomly sampled from D_s, where $y_i' \neq y_i$. For $x_i \in D_u$, contrastive sample x_i' can be randomly sampled from D_s.

So far we can get the contrastive sample sequence $X_i = \{x_{i1}, \ldots, x_{i4}\}$. The similarities between these contrastive and original samples differ from strong to weak.

4.2. Trigger Recognition

The trigger is specific information that leads to the occurrence of the event in event mention. Capturing trigger information helps the model better discern event types, making its learning and reasoning processes more accurate and meaningful. The existing zero-shot event detection method shows that the best representation of the event should contain the trigger information [5,8]. Inspired by ZEOP [15], we design a trigger recognition TR-prompt, which can obtain trigger representation.

TR-Prompt

As shown in Figure 3, TR-prompt is designed as "This is an event about `[MASK]`.<event mention>.", where <event mention> is the event text, and `[MASK]` is the trigger representation that BERT needs to predict. Now for each event instance, the input sequence of BERT is $w = \{w_{cls}, w_1, w_2, \ldots, w_M, w_{i+1}, \ldots, w_k, \ldots, w_n\}$, where w_i is the ith token of the template sentence, w_M is the `[MASK]` token and $\{w_{i+1}, \ldots, w_k, \ldots, w_n\}$ is event text token sequence. The eigenvector corresponding to `[MASK]` is h_M:

$$H = [h_{CLS}, \ldots, h_M, \ldots, h_n] = BERT(w). \tag{1}$$

Then, the feature vector h_M passes through the prediction layer and activation function based on the feedforward neural network to obtain the context representation of the trigger word:

$$h_M = \sigma(W_p h_M + b_p), \tag{2}$$

where W_p and b_p are the parameters of the prediction layer, and they are also part of the pretrained language model BERT. It should be noted that the triggers must be included in the event description text, and the BERT vocabulary as a solution space is far beyond this category, so we also need to define a conversion function. Specifically, this transformation function needs to adjust the predicted value of words that are not included in the event description text to 0:

$$h_o = \sigma(M \cdot W_p h_M + b_p), \tag{3}$$

where M is a mask that adjusts the predicted value of the word not included in the event description text to 0, and h_o is the final predicted trigger representation vector.

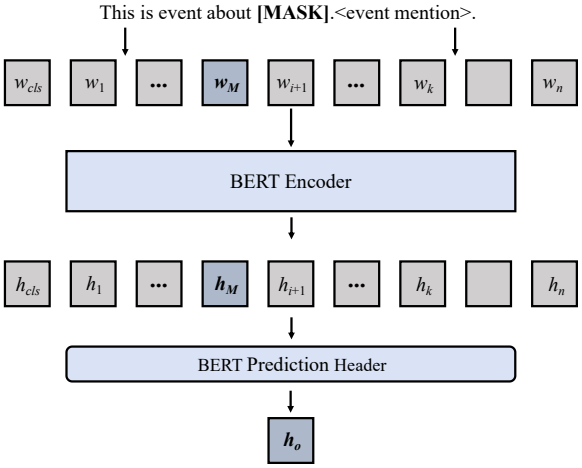


Figure 3. Trigger representations are obtained based on the TR-prompt. The trigger representation is ultimately obtained from the trigger probability distribution predicted by the BERT prediction header for the [MASK].

4.3. Event Sentence Embedding

The event sentences not only contain trigger information but also include event arguments and prepositions that reflect the relationship between events and arguments. Fully mining information from event instances to generate better sentence embeddings can better assist models in identifying different types of events in zero-shot scenarios. Due to the embedding bias and ineffective layers when the native BERT encodes unseen types, we also design a sentence embedding template ER-prompt following PromptBERT [16], which effectively uses the knowledge in each layer of BERT to directly predict the sentence embeddings.

ER-Prompt

Different from the traditional classification and QA tasks, the purpose of the ER-prompt is to directly obtain sentence embeddings of events by leveraging the predictive capability of the PLM. To assist the model in fully understanding event texts, we have designed an event-specific prompt template "This event: '<event mention>' means `[MASK]`.", as shown in Figure 4. Here, <event mention> represents the event text, and `[MASK]` is the sentence embedding that BERT needs to predict. ER-prompt generates two sentence embeddings after input into BERT. The sentence embedding h_p generated from the `[MASK]` part is predicted by BERT, capturing the overall event information. Like the TR-prompt, we can use Equation (2) to obtain h_p. The <event mention> part generates the sentence embedding h_n directly encoded by BERT, which does not contain any information about the event. To eliminate the influence of the template itself as much as possible, we use the mean pooling of the word vectors of the <event mention> part but not `[CLS]` marker to obtain h_n:

$$h_n = mean\ \ pooling([h_{i+1}, \ldots, h_n]). \tag{4}$$

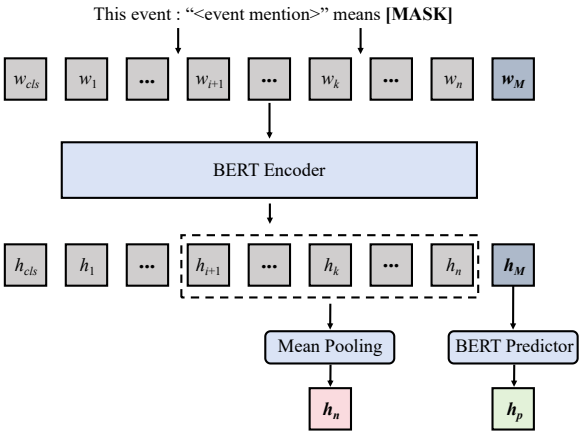


Figure 4. Event sentence embeddings are obtained based on the ER-prompt. In this method, the sentence embeddings result for the `[MASK]` part is directly obtained from the predicted event, while the sentence embeddings result for the <event mention> part, which represents the event text, is obtained through direct encoding by BERT.

4.4. Contrastive Fusion

At this point, we obtain vectors from three different perspectives of information. h_o is trigger representation, which are task-specific vectors, capturing information about the trigger in the events. h_p denotes event sentence embeddings, which are event-specific vectors, representing the overall information of the events. h_n refers to sentence embeddings directly encoded by BERT for the events, which are non-specific vectors and do not contain any specific information about the events.

Shallow concatenation fusion fails to capture the complex interaction between trigger representations and sentence embeddings. Therefore, we seek a method to capture the intricate interaction of merging these two types of information in the vector space. Contrastive fusion provides us with a promising approach, as illustrated in Figure 5.

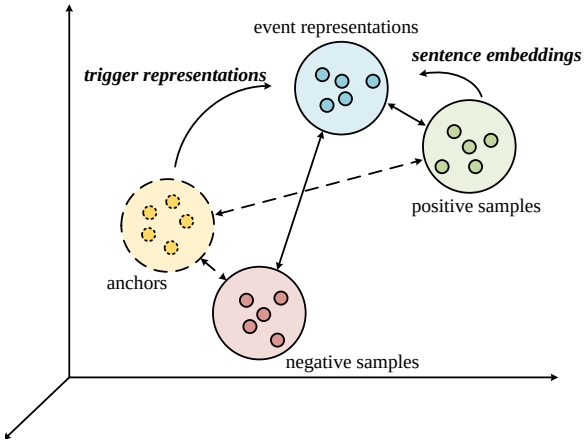


Figure 5. Contrastive fusion for trigger representations and sentence embeddings. Both interact and fuse in the vector space to obtain the final event representations.

In contrastive fusion, using h_o as the anchor, h_p is set as the positive example, and h_n as the negative example. Our goal is to minimize the distance between the anchor and the positive example while maximizing the distance from the negative example, facilitating a comprehensive interaction between trigger representations and sentence embeddings, while simultaneously reducing the influence of irrelevant event information. Considering that the feature vector encoded by the BERT pretrained language model is a dense vector with 768 dimensions, this paper chooses the Euclidean distance as the distance function.

Since only using target to define the loss will make the whole network difficult to converge, this paper uses the margin loss function to define the contrastive fusion loss:

$$L_m = max(0, margin - (d_p - d_n)), \tag{5}$$

where d_p represents the Euclidean distance between h_o and h_p, and d_n represents the Euclidean distance between h_o and h_n. Only when the difference between the two distances d_p and d_n is greater than the preset threshold, the value of the loss function is 0; otherwise, the smaller the difference between the two, the greater the value of the loss function.

So far, we have the final event representation $e = \{e_i, e_{i1}, \ldots, e_{i4}\}$ of the original event mention and the comparison sample sequence obtained in Section 4.1.

4.5. Event Type Prediction

Considering that zero-shot event detection contains seen and unseen events, for seen events, this type prediction directly corresponds to its real type, and the model can be directly used in traditional event detection tasks.

For unseen events, this type of prediction has no one-to-one correspondence with its true type, and can only be used as a clustering result. Based on this, we introduce the prototype network [28] to complete the event type prediction. 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, h is the dimension of event representation vector encoded by BERT. $n = s + u$ is the number of event types, where s is the number of seen types and u is the number of unseen types. After the prototype definition is completed, given the feature vector of event representation e, its probability of belonging to the ith types is

$$p(y = i|e) = \frac{exp(-d(e, c_i))}{\sum_{i'} exp(-d(e, c_{i'}))}, \tag{6}$$

where $d(e, c_i)$ is the Euclidean distance between event representation e and event prototype c_i. As shown in Figure 6, the semi-supervised algorithm based on prototype networks is explained from the perspective of the feature space regarding its role in enhancing model accuracy.

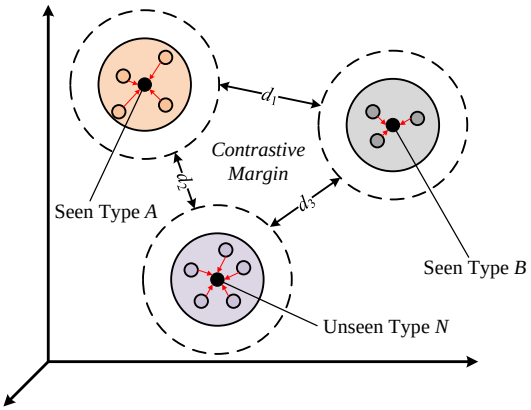


Figure 6. The prototypes defined by the prototype network, serving as type centers, can provide a reference standard for both the supervised loss function and the ordered pairwise loss function. This, in turn, assists the model in mapping samples to positions in the feature space that are closer together.

4.6. Loss Analysis

Ordered contrastive loss. The design goal of the ordered contrastive loss function is to maintain the order of the similarity between the four comparison samples and the original samples proposed in Section 4.1 After applying the prototype network, both the original samples and the contrastive samples in the ordered contrastive learning model can be represented by the probability distribution on $n = s + u$ event types, instead of the

high-dimensional dense vector output by the event encoder. Therefore, we use Wasserstein distance [29] rather than Euclidean distance as the distance metric function in the ordered contrastive loss:

$$d_w(\mu, v) = \left(\inf_{\gamma \in \Gamma(\mu,v)} \int_{X \times Y} d^p(x,y) d\gamma(x,y) \right)^{\frac{1}{p}}, \tag{7}$$

where μ and v represent two probability distributions to be compared, respectively; p is the order, and 1 is taken in this paper. The distance from the four contrastive samples to the original sample d_1, d_2, d_3, d_4 is obtained. In order to maintain the partial order relationship of the similarity of ordered comparison samples, that is, for the seen type $d_1 < d_2 < d_3 < d_4$, for the unseen type $d_1 < d_2 < d_3 \neq d_4$, the final ordered contrastive loss function is defined as

$$L_c = d_1 + L_m(d_2, d_1) + L_m(d_3, d_2) + \begin{cases} L_m(d_3, d_4) & x \in S, \\ L_m(d_4, d_2) & x \in U. \end{cases} \tag{8}$$

Contrastive fusion loss. In Section 4.4 an implementation method for the contrastive fusion module is presented, where the distances between positive and negative samples and the anchor are denoted as d_p and d_n using the Euclidean distance. Therefore, the loss function for event contrast representation is defined as

$$L_f = max(0, margin - (d_p - d_n)). \tag{9}$$

Event type prediction loss. The core idea of the semi-supervised zero-shot event detection algorithm is to apply different loss functions to seen event type samples and unseen event type samples. The semi-supervised loss function used in the SS-VQ-VAE model proposed by Huang and Ji [5] is

$$loss = \sum_{x \in S} -\widehat{y} \cdot log(y) + \sum_{x \in U} max(y^{1:k}) - max(y^{k+1:n}), \tag{10}$$

where $\widehat{y}$ represents the true event type label, while y represents the predicted event type given by the model. The design of this loss function can separate unseen event samples from known event types, it is still challenging to differentiate between different unseen event types. Additionally, the assumption that the model can guarantee all unlabeled samples belong to unseen event types, which is implied by this contrastive loss function, is difficult to satisfy in practice. To address these issues, we have improved the previous work. For unseen event types, the loss for event type is directly defined as 0. Therefore, the event type loss function defined for sample x in this paper is as

$$L_e = \begin{cases} -\widehat{y} log(y_x), & x \in S, \\ 0, & x \in U, \end{cases} \tag{11}$$

Trigger prediction loss. The event classification algorithm based on prototype networks leverages annotated information of known events to provide supervisory signals for labeled samples. By calculating the negative logarithm loss function, the algorithm aids in the learning of type-specific features of known events by the model. The utilization of trigger word predictions provided by TR-prompt in the previous context enables the usage of supervised signals based on trigger words, thereby assisting the model in better accomplishing the task of trigger word recognition. Therefore, this paper proposes a newly defined negative logarithm loss function based on trigger word supervisory signals:

$$L_t = \begin{cases} -\widehat{t} log(t_x), & x \in S, \\ 0, & x \in U, \end{cases} \tag{12}$$

Equations (8), (9), (11), and (12) are combined to form the final loss function of the model:

$$Loss = L_c + L_f + L_e + L_t. \tag{13}$$

5. Experiments

5.1. Implementation Details

We implement all the algorithms in the experiment using Python packages PyTorch and the Transformer Library. For all algorithms that require the use of the BERT pre-trained language model, the article consistently employs the *bert-base-uncased-model*. This model is equipped with 12 layers of multi-head self-attention encoders and features 768-dimensional vectors. The total parameter scale is approximately 109 million. To implement the back-translation method for generating rephrased samples in Section 4.1, we utilize the machine translation model provided by the Argos Translate library (https://www.argosopentech.com, accessed on 1 August 2023), setting Chinese as the intermediate language for translation. For training deep neural network models, AdamW is used as the optimizer to iterate through model parameters. The learning rate for the BERT model parameters undergoes a grid search within the range of $[1 \times 10^{-7}, 1 \times 10^{-4}]$, while for non-BERT models, the learning rate is searched within $[1 \times 10^{-4}, 1 \times 10^{-2}]$. All experiments in this article were conducted on a Linux server equipped with two RTX 3090 GPUs.

5.2. Datasets

We select two publicly available event detection datasets both contain English event description texts from the general domain. `ACE-2005` (https://catalog.ldc.upenn.edu/LDC2006T06, accessed on 1 August 2023) is a classic event detection dataset widely used in event detection research. `FewShotED` [30] is a relatively new dataset that introduces the concept of seen and unseen event types to simulate imbalances in the number of samples for different event types.

To balance the number of samples between seen and unseen event types, we adopted the same data partitioning strategy as in the previous work [15]. Sort all types of events by their quantity, where event types at odd positions are labeled as seen types, while event types at even positions are labeled as unseen types.

We randomly divide the dataset into training, validation, and test sets in an 8:1:1 ratio. These detail steps and strategies for dataset configuration ensured the reproducibility of the experiments and provided a benchmark for evaluating model performance. Additionally, the introduction of the `FewShotED` dataset allows us to conduct zero-shot event detection experiments, enabling a more comprehensive assessment of the model's generalization capabilities. The statistics of the processed dataset are shown in Table 1.

Table 1. The statistical information of the `ACE-2005` and `FewShotED` datasets, where E represents the number of event types, N represents the number of samples, and T represents the number of trigger words.

Dataset	`ACE-2005`			`FewShotED`		
	$\|E\|$	$\|N\|$	$\|T\|$	$\|E\|$	$\|N\|$	$\|T\|$
Seen	17	2316	565	50	40,893	1565
Unseen	16	1489	463	50	33,439	1907
Total	33	3805	1028	100	74,332	3472
Mean		115.30			743.32	
Stdev		206.32			2828.47	

5.3. Evaluation

Our experiments are designed around two tasks: firstly, a conventional event detection task, which is essentially a classification task when considering the seen event samples in

the datasets. Secondly, a zero-shot event detection task for unseen event samples in the datasets, which formally resembles a clustering task. The classic classification metric, F1 score, will serve as the common evaluation metric for both tasks:

$$F1 = 2 \times \frac{p \times r}{p + r}, \tag{14}$$

where p represents *Precision*, and r represents *Recall*:

$$p = \frac{TP}{TP + FP}, \tag{15}$$

$$r = \frac{TP}{TP + FN}, \tag{16}$$

where TP represents the number of true positives, FP represents the number of false positives, and FN represents the number of false negatives. For seen events, the predicted labels used in F1 score computation are directly provided by the model. For unseen event types, a mapping is established first using the Hungarian algorithm before calculating the F1 score. Additionally, taking inspiration from the works of SS-VQ-VAE and SCCL [5,31], we also employ normalized mutual information (NMI) and Fowlkes–Mallows score (FM) to assess clustering performance for unknown event types. NMI score is the normalized result of mutual information score, used to measure the degree of containment of one set within another set:

$$NMI(X, Y) = \frac{2 \times I(X; Y)}{[H(X) + H(Y)]}, \tag{17}$$

where X represents the true labels of the samples, Y represents the predicted labels; $H(X)$ is the entropy function, and $I(X; Y)$ is the mutual information function. FM score is a metric used to evaluate the similarity between two clusters:

$$FM(X, Y) = \frac{TP}{\sqrt{(TP + FP) \times (TP + FN)}}. \tag{18}$$

5.4. Baseline

In order to compare our proposed model with existing methods in the field of zero-shot event detection, we chose the following models as baseline models:

- **Supporting Clustering with Contrastive Learning (SCCL)** is one of the best-performing models in unsupervised text clustering tasks, achieving text clustering by optimizing a top-down clustering loss. SCCL is used to detect new event types based on unseen event mentions. The contextual feature of trigger tokens are used in our experiments [31].
- **The Semi-supervised Vector Quantized Variational Autoeocoder (SS-VQ-VAE)** is a semi-supervised zero-shot event detection model that also utilizes BERT as the encoder for event text. It employs a variational autoencoder to learn discrete event features. SS-VQ-VAE is trained based on visible event types and annotations, and it can be applied to zero-shot event detection [5].
- **BERT Ordered Contrastive Learning (BERT-OCL)** designs an ordered contrastive learning method for clustering unseen event types. The Euclidean distance is used to compute pairwise distance between examples for reducing intra-class distances and increasing inter-class distances [15,32].
- **Zero-Shot Event Detection with Ordered Contrastive Learning (ZEOP)** leverages prompt learning and ordered contrastive loss based on both instance-level and class-level distance for zero-shot event detection. ZEOP identifies trigger tokens then predicts event types by clustering [15].

- **APEX Prompt** is based on prompt engineering. APEX Prompt utilizes a more comprehensive event type description as a template. Compared to other prompt-based methods, this method can significantly enhance the performance of event detection, especially in zero-shot event detection [33].

For the performance of the above baseline models, the APEX Prompt results on the ACE-2005 dataset are taken from Wang et al. [33], the results on FewShotED are provided by our experiments, and other results are directly taken from Zhang et al. [15].

6. Results and Analysis

6.1. Main Result

The main result of our model and all baseline approaches are shown in Table 2, which is divided into two parts by the used datasets. From the results in the table, the following observations can be made: (1) Our proposed model achieved the best overall performance on both datasets, whether it is seen or unseen event detection tasks. (2) For both baseline models and our model, the performance on FewShotED is relatively higher than on ACE-2005 (the F1 scores for the seen and unseen are higher by 15.62% and 9.68% respectively). This may be attributed to the larger number of samples included in FewShotEvent, which helps the model learn better event representations. (3) The performance improvement of our model on unseen event detection tasks is significantly higher than on seen event detection tasks (the improvement on the unseen in ACE-2005 is 10.79% higher than the improvement on the seen, and it is 7.97% higher in FewShotED). This indicates the effectiveness of our model in detecting unseen events.

Table 2. The overall results. The data marked in **bold** are the results by our model COPE, and the data marked with an underline are the best results of the competing baselines.

Model	ACE-2005			
	F1-Seen	**F1-Unseen**	**NMI**	**FM**
SCCL	0.5999	0.3190	0.3259	0.2403
SS-VQ-VAE	0.6988	0.3509	0.2515	0.4269
BERT-OCL	0.6040	0.3751	0.4532	0.2551
ZEOP	0.7771	0.4591	0.3797	0.4913
APEX Prompt	0.7490	0.5530	-	-
COPE	**0.7904**	**0.5803**	**0.4952**	**0.5097**
Model	FewShotED			
	F1-Seen	**F1-Unseen**	**NMI**	**FM**
SCCL	0.8717	0.3640	0.2647	0.3462
SS-VQ-VAE	0.9208	0.4364	0.1722	0.5762
BERT-OCL	0.9017	0.2160	0.4157	0.1894
ZEOP	0.9306	0.5814	0.4831	0.7139
APEX Prompt	0.9327	0.6371	-	-
COPE	**0.9466**	**0.6771**	**0.5392**	**0.7298**

Overall, the prompt for representation contrastive learning model we proposed is particularly effective in learning distinct representations for different types of events, thereby enhancing the performance of zero-shot event detection.

6.2. Ablation Analysis

To investigate the impact of different modules on the model's performance, we conduct ablation experiments by removing ordered contrastive learning, retaining only trigger recognition, retaining only sentence embedding, and removing contrastive fusion from

the model. In the experiment where only trigger recognition is retained, we use trigger representation as the final event representations. In the experiment where only sentence embedding is retained, we use sentence embeddings generated by the `[MASK]` part of the ER-prompt as the final event representation. When contrastive fusion is removed, concatenation of trigger representation and sentence embedding is used as the final event representation. The performance comparison of these ablations is shown in Table 3.

Table 3. The results of ablation experiments. "w/o" indicates that the referred part of the model was removed in experiments. The data marked in **bold** are the best experiment results among the models.

Model	ACE-2005			
	F1-Seen	**F1-Unseen**	**NMI**	**FM**
COPE	0.7904	**0.5803**	**0.4952**	**0.5097**
w/o OCL	**0.8166**	0.5537	0.3494	0.4316
only trigger recognition	0.7587	0.4305	0.4592	0.5138
only sentence embedding	0.7362	0.5544	0.3129	0.3351
w/o contrastive fusion	0.7803	0.5695	0.3797	0.4913
Model	**FewShotED**			
	F1-Seen	**F1-Unseen**	**NMI**	**FM**
COPE	0.9466	**0.6771**	**0.5392**	**0.7298**
w/o OCL	**0.9581**	0.6351	0.4783	0.5493
only trigger recognition	0.9207	0.5668	0.5147	0.6816
only sentence embedding	0.9019	0.6122	0.4659	0.7246
w/o contrastive fusion	0.9389	0.6503	0.4831	0.7139

From the results in the table, we observe that removing different modules from the model has an impact on its performance. When the ordered contrastive learning is removed, we see a decrease in the F1 scores for unseen types by 2.66% and 4.2% on `ACE-2005` and `FewShotED`, respectively. This suggests that ordered contrastive learning indeed helps the model learn better feature representations for unseen types. However, we also find that in this case, the F1 scores for seen types on both datasets improve by 2.62% and 1.15%, respectively. This indicates that ordered contrastive learning does contribute to the model's ability to learn better feature representations for unseen types, but this benefit comes at a cost. The introduction of ordered contrastive sample sequences inevitably introduces interference to the model, affecting the model's recognition of labels for seen types as its supervisory signals, leading to conflicts with the training objectives of supervised learning, and thereby impairing some performance.

The impact of retaining only trigger word recognition is significantly greater for unseen types than for seen types. The F1 scores for unseen types on both datasets decrease by 14.98% and 11.03%, while the F1 scores for seen types decrease by 3.17% and 2.59%, respectively. This indicates that the introduction of background sentence embeddings enhances the model's ability to handle unseen types more effectively, emphasizing the importance of task-agnostic information in event instances for zero-shot ED. Additionally, introducing prompts in sentence encoding effectively addresses the issues present in the original PLM.

The impact of retaining only sentence embeddings is more significant for seen types than for unseen types, with a 2.83% larger decrease in F1 scores for seen types than for unseen types in `ACE-2005` and a 2.02% larger decrease in `FewShotED`. For event detection tasks, trigger word recognition is crucial in distinguishing them from text classification tasks. When key words triggering events cannot be identified, the model loses specific information about the event types. In such a situation, the model may be more inclined to use other features or contextual information in the text to accomplish tasks, such as

utilizing semantic and structural information in the text. This makes the model more likely to perform text classification and clustering tasks. For seen types, the supervision signal is weakened, and the model is likely to capture features unrelated to event types, making it more challenging to identify event types.

Removing the contrastive fusion also resulted in a decline in the overall performance of the model. Compared to ZEOP, although the model's performance improved with enhanced sentence embeddings, the model's performance is not ideal due to the shallow fusion strategy of only using concatenation fusion, which lead to the omission of complex interaction information between trigger representation and sentence embedding. After applying contrastive fusion, the final event representation integrated trigger information and overall event information, capturing the interaction information between them adequately, thus enhancing the model's performance.

6.3. Prompt Effect

For the prompt-based component of the model, similar to traditional generative tasks, template selection is crucial for the results. We categorize templates into two types: relationship tokens mainly represent the relationship between <event mention> and `[MASK]`, while prefix tokens enclose <event mention> and incorporate some semantic information. We manually design several templates based on the work of Jiang et al. [16] and compare their performance, as shown in Table 4.

Table 4. The impact of different templates on model performance. "This event: '<event mention>' means `[MASK]`" is the template used in our model. The **bold** indicates the best model performance under this template.

Template	ACE-2005		FewShotED	
	F1-Seen	**F1-Unseen**	**F1-Seen**	**F1-Unseen**
	Searching for relationship tokens			
<event mention>`[MASK]`.	0.5641	0.2402	0.7741	0.3508
<event mention> is `[MASK]`.	0.6013	0.3194	0.7831	0.4296
<event mention> mean `[MASK]`.	0.6223	0.3862	0.7826	0.4996
<event mention> means `[MASK]`.	0.6959	0.4824	0.7988	0.5368
	Searching for prefix tokens			
This <event mention> means `[MASK]`.	0.7670	0.4887	0.9235	0.6138
This event of <event mention> means `[MASK]`.	0.7766	0.5365	0.9291	0.6355
This event of "<event mention>" means `[MASK]`.	0.7791	0.5487	0.9324	0.6426
This event: "<event mention>" means `[MASK]`.	**0.7953**	**0.5812**	**0.9418**	**0.6828**

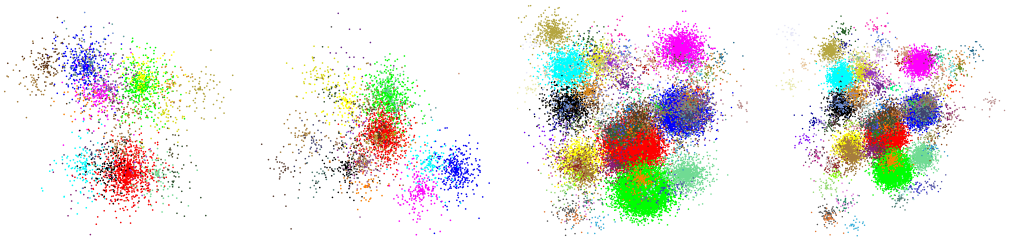
Overall, it can be observed that complex prefix templates containing partial semantic information outperform simple concatenation with `[MASK]` and event mentions in terms of relationship templates. These complex prefix templates lead to improved model performance on both datasets.

Furthermore, the improvements brought about by modifying templates on both datasets are primarily focused on the performance enhancement of unseen types. This is mainly because prompt-based methods typically do not rely on domain-specific knowledge or annotated data, thus exhibiting a degree of generality across different domains and tasks. This flexibility allows these methods to adapt easily to various application areas without the need for retraining or fine-tuning the model. Prompt-based approaches can leverage unsupervised data, effectively harnessing knowledge from various layers of pretrained language models, thereby enhancing model performance and generalization capability.

6.4. Qualitative Analysis

We additionally present qualitative analysis of our zero-shot results by comparing the feature visualization of event representation [20].

On two different datasets, we compare the event representations obtained by directly encoding them with BERT and the event representations obtained using our encoding method. The results in Figure 7 show that encoding events directly with the original BERT leads to overlapping event representations, making it difficult to separate events of different categories in the vector space. Additionally, we observe that events with fewer samples tend to mix with events with more samples after being encoded by BERT, which is consistent with the embedding bias issue introduced by BERT.



(**a**) BERT for ACE-2005 (**b**) COPE for ACE-2005 (**c**) BERT for FewShotED (**d**) COPE for FewShotED

Figure 7. Feature visualization of event representation on ACE-2005 and FewShotED. The four images respectively represent the results obtained by directly encoding two datasets with BERT and the representations obtained by encoding events using the method proposed in this paper.

After using our model's encoding method, we can clearly observe that event representations with fewer samples are effectively separated from those with more samples, with the most notable separation occurring in the unseen type of events. Taking ACE-2005 as an example, the three most numerous event types in the unseen types are *Movement:Transport*; *Contact:Meet* and *Transaction:Transfer-Money* (represented in cyan, pink, and yellow, respectively). After applying our method, these types are well separated, and event types with fewer instances, such as *Personnel:Nominate* and *Justice:Pardon*, are no longer mixed with the representations of events with more instances. Instead, they are relatively scattered in other areas of the vector space, which is advantageous for the model to better distinguish between event types.

7. Limitations

While COPE has shown promising results in zero-shot ED, experiments indicate that there is still room for improvement in the algorithm, with potential for further exploration in the future. The introduction of ordered contrastive learning has indeed enhanced the detection of unseen events, but it has also introduced certain conflicts with the supervised learning task of detecting seen events. In the prototype network, prototypes play the role of type centers. While random initialization is convenient operationally, it inevitably reduces the model fitting speed, and the selection of hyperparameters for the number of unseen types also has an impact. Additionally, in our dataset partitioning, seen and unseen types are evenly distributed, whereas in real-world scenarios, the distribution of these two types of data may not be equal. Improving the model to make it more adaptable to a wider range of scenarios is a challenge to be addressed in the future.

8. Conclusions

This paper introduces COPE, a dual-contrastive learning model designed to effectively tackle the zero-shot event detection challenge. It presents innovative perspectives and strategies for event representation learning through the utilization of two levels of contrastive learning: ordered contrastive learning and contrastive fusion learning. At the sample level, our approach enables precise differentiation between diverse events, while at the instance level, we achieve a harmonious integration of task-specific and task-agnostic embeddings. In our experiments, we comprehensively evaluate COPE using two widely-used event detection datasets. The results highlight the outstanding performance of our method in zero-shot ED, significantly enhancing both accuracy and semantic information

capture. By intricately modeling the interplay between event triggers and background sentences, our approach adeptly captures nuanced event features within high-dimensional vector space, thereby elevating zero-shot ED capabilities.

Author Contributions: Conceptualization, J.L. and P.H.; methodology, J.L. and B.G.; formal analysis and investigation, H.X. and H.H.; writing—original draft preparation, J.L.; writing—review and editing, B.G. and H.X.; supervision, B.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The ACE-2005 dataset is available at https://catalog.ldc.upenn.edu/LDC2006T06 (accessed on 1 August 2023). The FewShotED dataset can be obtained from the GitHub project: https://github.com/231sm/Low_Resource_KBP (accessed on 1 August 2023).

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

References

1. Nguyen, T.H.; Grishman, R. Graph Convolutional Networks with Argument-Aware Pooling for Event Detection. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th Innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LO, USA, 2–7 February 2018; McIlraith, S.A., Weinberger, K.Q., Eds.; PKP: Burnaby, BC, Canada, 2018; pp. 5900–5907. [CrossRef]
2. Wadden, D.; Wennberg, U.; Luan, Y.; Hajishirzi, H. 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), Hong Kong, China, 3–7 November 2019; pp. 5784–5789. [CrossRef]
3. Lin, Y.; Ji, H.; Huang, F.; Wu, L. A Joint Neural Model for Information Extraction with Global Features. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 7999–8009. [CrossRef]
4. Zhang, C.; Soderland, S.; Weld, D.S. Exploiting Parallel News Streams for Unsupervised Event Extraction. *Trans. Assoc. Comput. Linguist.* **2015**, *3*, 117–129. [CrossRef]
5. Huang, L.; Ji, H. Semi-supervised New Event Type Induction and Event Detection. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 718–724. [CrossRef]
6. Wang, Z.; Wang, X.; Han, X.; Lin, Y.; Hou, L.; Liu, Z.; Li, P.; Li, J.; Zhou, J. 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), Online, 1–6 August 2021; pp. 6283–6297. [CrossRef]
7. Huang, L.; Ji, H.; Cho, K.; Dagan, I.; Riedel, S.; Voss, C.R. Zero-Shot Transfer Learning for Event Extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, VIC, Australia, 15–20 July 2018; Gurevych, I., Miyao, Y., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2018; Volume 1: Long Papers, pp. 2160–2170. [CrossRef]
8. Zhang, H.; Wang, H.; Roth, D. Zero-shot Label-Aware Event Trigger and Argument Classification. In Proceedings of the Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online, 1–6 August 2021; pp. 1331–1340. [CrossRef]
9. Lyu, Q.; Zhang, H.; Sulem, E.; Roth, D. 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), Online, 1–6 August 2021; pp. 322–332. [CrossRef]
10. Lee, I.T.; Goldwasser, D. Multi-Relational Script Learning for Discourse Relations. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; Korhonen, A., Traum, D., Màrquez, L., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2019; pp. 4214–4226. [CrossRef]
11. Rezaee, M.; Ferraro, F. Event Representation with Sequential, Semi-Supervised Discrete Variables. In Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Seattle, WA, USA, 6–11 June 2021; Toutanova, K., Rumshisky, A., Zettlemoyer, L., Hakkani-Tur, D., Beltagy, I., Bethard, S., Cotterell, R., Chakraborty, T., Zhou, Y., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2021; pp. 4701–4716. [CrossRef]
12. Deng, S.; Zhang, N.; Li, L.; Hui, C.; Huaixiao, T.; Chen, M.; Huang, F.; Chen, H. OntoED: Low-resource Event Detection with Ontology Embedding. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Online, 1–6 August; Zong, C., Xia, F., Li, W., Navigli, R., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2021; pp. 2828–2839. [CrossRef]

13. Martin, L.J.; Ammanabrolu, P.; Wang, X.; Hancock, W.; Singh, S.; Harrison, B.; Riedl, M.O. Event Representations for Automated Story Generation with Deep Neural Nets. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LO, USA, 2–7 February 2018; McIlraith, S.A., Weinberger, K.Q., Eds.; PKP: Burnaby, BC, Canada, 2018; pp. 868–875. [CrossRef]
14. Chen, H.; Shu, R.; Takamura, H.; Nakayama, H. GraphPlan: Story Generation by Planning with Event Graph. In Proceedings of the 14th International Conference on Natural Language Generation, Aberdeen, UK, 20–24 September 2021; Belz, A., Fan, A., Reiter, E., Sripada, Y., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2021; pp. 377–386. [CrossRef]
15. Zhang, S.; Ji, T.; Ji, W.; Wang, X. Zero-Shot Event Detection Based on Ordered Contrastive Learning and Prompt-Based Prediction. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, USA, 10–15 July 2022; pp. 2572–2580. [CrossRef]
16. Jiang, T.; Jiao, J.; Huang, S.; Zhang, Z.; Wang, D.; Zhuang, F.; Wei, F.; Huang, H.; Deng, D.; Zhang, Q. PromptBERT: Improving BERT Sentence Embeddings with Prompts. In Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 25–28 June 2022; pp. 8826–8837. [CrossRef]
17. Socher, R.; Chen, D.; Manning, C.D.; Ng, A. Reasoning with Neural Tensor Networks for Knowledge Base Completion. *Neural Inf. Process. Syst.* **2013**, *26*, 926–934.
18. Weber, N.; Balasubramanian, N.; Chambers, N. Event Representations with Tensor-Based Compositions. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, LO, USA, 2–7 February 2018; McIlraith, S.A., Weinberger, K.Q., Eds.; PKP: Burnaby, BC, Canada, 2018; pp. 4946–4953. [CrossRef]
19. Ding, X.; Liao, K.; Liu, T.; Li, Z.; Duan, J. Event Representation Learning Enhanced with External Commonsense Knowledge. 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), Hong Kong, China, 3–7 November 2019; Inui, K., Jiang, J., Ng, V., Wan, X., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2019; pp. 4894–4903. [CrossRef]
20. Gao, J.; Wang, W.; Yu, C.; Zhao, H.; Ng, W.; Xu, R. Improving Event Representation via Simultaneous Weakly Supervised Contrastive Learning and Clustering. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; Muresan, S., Nakov, P., Villavicencio, A., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2022; pp. 3036–3049. [CrossRef]
21. Gao, T.; Yao, X.; Chen, D. SimCSE: Simple Contrastive Learning of Sentence Embeddings. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Online, Punta Cana, Dominican Republic, 7–11 November 2021; pp. 6894–6910. [CrossRef]
22. Yan, Y.; Li, R.; Wang, S.; Zhang, F.; Wu, W.; Xu, W. ConSERT: A Contrastive Framework for Self-Supervised Sentence Representation Transfer. 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), Online, 1–6 August 2021; pp. 5065–5075. [CrossRef]
23. Zheng, J.; Cai, F.; Chen, H. Incorporating Scenario Knowledge into A Unified Fine-tuning Architecture for Event Representation. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2020, Virtual Event, China, 25–30 July 2020; Huang, J.X., Chang, Y., Cheng, X., Kamps, J., Murdock, V., Wen, J., Liu, Y., Eds.; Association for Computing Machinery: New York, NY, USA, 2020; pp. 249–258. [CrossRef]
24. Vijayaraghavan, P.; Roy, D. Lifelong Knowledge-Enriched Social Event Representation Learning. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume, Online, 19–23 April 2021; Merlo, P., Tiedemann, J., Tsarfaty, R., Eds.; Association for Computational Linguistics: Baltimore, MD, USA, 2021; pp. 3624–3635. [CrossRef]
25. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992. [CrossRef]
26. Li, B.; Zhou, H.; He, J.; Wang, M.; Yang, Y.; Li, L. On the Sentence Embeddings from Pre-trained Language Models. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 9119–9130. [CrossRef]
27. Logeswaran, L.; Lee, H. An efficient framework for learning sentence representations. *arXiv* **2018**, arxiv:1803.02893.
28. Snell, J.; Swersky, K.; Zemel, R.S. Prototypical Networks for Few-shot Learning. In Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, Long Beach, CA, USA, 4–9 December 2017; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H.M., Fergus, R., Vishwanathan, S.V.N., Garnett, R., Eds.; 2017; pp. 4077–4087. Available online: https://proceedings.neurips.cc/paper/2017/hash/cb8da6767461f2812ae4290eac7cbc42-Abstract.html (accessed on 25 April 2024).

29. Kolouri, S.; Nadjahi, K.; Simsekli, U.; Badeau, R.; Rohde, G.K. Generalized Sliced Wasserstein Distances. In Proceedings of the Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, Vancouver, BC, Canada, 8–14 December 2019; Wallach, H.M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E.B., Garnett, R., Eds.; 2019; pp. 261–272. Available online: https://proceedings.neurips.cc/paper/2019/hash/f0935e4cd5920aa6c7c996a5ee53a70f-Abstract.html (accessed on 25 April 2024).
30. Deng, S.; Zhang, N.; Kang, J.; Zhang, Y.; Zhang, W.; Chen, H. Meta-Learning with Dynamic-Memory-Based Prototypical Network for Few-Shot Event Detection. In Proceedings of the WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 February 2020; Caverlee, J., Hu, X.B., Lalmas, M., Wang, W., Eds.; Association for Computing Machinery: New York, NY, USA, 2020; pp. 151–159. [CrossRef]
31. Zhang, D.; Nan, F.; Wei, X.; Li, S.W.; Zhu, H.; McKeown, K.; Nallapati, R.; Arnold, A.O.; Xiang, B. 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, Online, 6–11 June 2021; pp. 5419–5430. [CrossRef]
32. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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), Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186. [CrossRef]
33. Wang, S.; Yu, M.; Huang, L. The Art of Prompting: Event Detection based on Type Specific Prompts. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Toronto, ON, Canada, 9–14 July 2023; pp. 1286–1299. [CrossRef]