Knowledge-Based Systems 265 (2023) 110375

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Multi-view entity type overdependency reduction for event argument extraction

Jing Xu [a], Dandan Song [a,*], Siu Cheung Hui [b], Fei Li [c], Hao Wang [a]

[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[b] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[c] Baidu Inc., China

ARTICLE INFO

Article history:
Received 30 May 2022
Received in revised form 3 February 2023
Accepted 6 February 2023
Available online 9 February 2023

Keywords:
Event argument extraction
Entity type overdependency
Feature representation
Contrastive learning

ABSTRACT

Event Argument Extraction (EAE) is a key component of event extraction, which has become a bottleneck that limits the overall performance of event extraction. As an entity-based extraction task, most EAE models focus on modeling complex interactions between entity mentions and event triggers. However, the strong correlation between entity types and argument role types has been overlooked in most EAE models, which disregard the possible negative effects of the correlation. In this paper, we study entity type dependency and conduct experiments to evaluate its effects on the overall performance for EAE. The experimental analysis shows that baseline EAE models suffer from varying degrees of entity type overdependency, which degrades the overall performance. To tackle this problem for EAE, we propose a novel multi-view entity type overdependency reduction model. The proposed model consists of two contrastive learning methods from different views and a cyclic training strategy. In particular, we propose a select-then-weigh contrastive learning method to achieve entity type overdependency reduction from the view of positive samples. And in parallel, we propose a pseudo-positive contrastive learning method to achieve entity type overdependency reduction from the view of negative samples. Moreover, the cyclic training strategy is designed to enable the two contrastive learning methods to collaborate efficiently. We have conducted experiments on the widely used ACE 2005 English dataset to evaluate the effectiveness of our proposed model. The experimental results show that our proposed model has outperformed the current state-of-the-art models for the EAE task.

1. Introduction

As a key component of event extraction, Event Argument Extraction (EAE) aims to find entity mentions participated in an event as event arguments and determine their corresponding roles. We take the sentence "*Tom drives back to Boston passing through Chicago*" as an example. Given that the event trigger is "*drives*" and its corresponding event type is "*Transport*", an EAE system aims to recognize the entity mention "*Boston*" as an event argument and "*Destination*" as its role type. As an entity-based extraction task, most EAE models focus on how to model the complex interactions between entity mentions and event trigger words, given the contextual semantic information [1–3]. From these models, we observe the following two data characteristics in most EAE datasets such as the ACE 2005 English dataset. First, entity types provide the main source of information for entity mentions for EAE. For example, following the similar setting on relation extraction in [4], we replace entity mentions by their entity types in the ACE 2005 English dataset. We then train DMBERT [5], which is an efficient and representative EAE model, for EAE. Results show that DMBERT achieves almost the same performance results before and after the replacement. Second, most entity types participate only in a few specific role types in EAE [6]. For example, 43 entity types appear in the ACE 2005 English dataset, but over 74% entity types only participate in less than 5 role types.

As observed from the above two data characteristics, we find that entity type information is vital for EAE and correlates strongly with role types. Current EAE models take advantage of entity type information [6–8] due to its significance in EAE but are not aware of the possible effects of the correlation between entity types and role types. In the example sentence "*A rocket holding the first of two Mars rovers blasted off Tuesday on a seven-month voyage to the planet.*" given in Table 1, the event type "*Transport*" is triggered by "*voyage*". An EAE model should recognize the entity mention "*two Mars rovers*" with the entity type "*Landing*", which does not participate in any role (i.e., "None"). However, as

* Corresponding author.
E-mail addresses: xujing@bit.edu.cn (J. Xu), sdd@bit.edu.cn (D. Song), ASSCHUI@ntu.edu.sg (S.C. Hui), lifei21@baidu.com (F. Li), wanghaobit@bit.edu.cn (H. Wang).

https://doi.org/10.1016/j.knosys.2023.110375
0950-7051/

Table 1
An example of entity type overdependency. The trigger words and event argument candidates are in red and blue, respectively.

Instances	Entity type	Golden role type	Predicted role type
Example: A rocket holding the first of two Mars rovers blasted off Tuesday on a seven-month voyage to the planet.	Landing	None	Vehicle ×
S1: The rovers' landing sites, on opposite sides of the planet, were chosen for their likelihood of holding evidence of water.	Landing	Vehicle	Vehicle ✓
S2: The bus was ripped to shreds while traveling between a residential area and Haifa university.	Landing	Vehicle	Vehicle ✓

most other entity mentions with entity type "*Landing*" participate in the role type "*Vehicle*" in the EAE dataset (e.g., S1 and S2 in Table 1), most EAE models are misled to wrongly recognize the entity mention "*two Mars rovers*" as participating in the role type "*Vehicle*". In this paper, we define such effect as entity type overdependency and conduct an experiment to evaluate whether it will hinder EAE models from understanding the semantic information from texts when predicting the correct role types. From the experiment, we find that entity type overdependency has degraded the overall performance of different baseline models for EAE. Intuitively, if we are able to reduce the degree of entity type overdependency when training EAE models, their performance should be improved.

As entity type overdependency is caused by heavy reliance on entity types for EAE, we should consider enabling EAE models to learn more semantic information besides entity types. Contrastive learning, which is widely used in self-supervised learning for computer vision [9,10] and natural language processing [11,12], is an effective approach for tackling this problem. In particular, we propose to use supervised contrastive learning [13–15], which pulls feature representations of instances belonging to the same type (called positive samples) together and pushes apart feature representations of instances whose types are different (called negative samples). However, vanilla supervised contrastive learning [13] can only learn role type information but not consider entity type information at the same time. It is a challenging problem on how to incorporate entity type information in contrastive learning for tackling entity type overdependency effectively.

In this paper, we propose a novel Multi-view Entity Type Overdependency Reduction (METOR) model to tackle the entity type overdependency problem. Specifically, two entity type overdependency reduction methods are proposed from different views, namely positive samples and negative samples. Regarding positive samples, we propose a select-then-weigh contrastive learning method to increase the similarity between the learned feature representations of instances with the same role type but having different entity types. Similarly, for negative samples, we propose a pseudo-positive contrastive learning method to reduce the similarity between the learned feature representations of instances with different role types but having the same entity type. Finally, to enable the collaboration between the two contrastive learning methods, we propose a cyclic training strategy for training the two methods more efficiently.

The main contributions of this paper are as follows:

- We conduct experiments to show that different kinds of EAE models are suffering from varying degrees of entity type overdependency, which degrades the overall performance. To the best of our knowledge, this study is the first to exploit the negative effects of overdependency on entity types for EAE.
- We propose a novel Multi-view Entity Type Overdependency Reduction (METOR) model, which consists of two novel contrastive learning methods and a cyclic training strategy, to tackle the entity type overdependency problem. The cyclic training strategy enables efficient collaboration between the two contrastive learning methods.
- Extensive experiments are conducted on the widely used ACE 2005 English corpus dataset. Our proposed METOR model has outperformed the different EAE baseline models. More specifically, the proposed METOR model achieves the state-of-the-art performance for the EAE task.

The rest of this paper is organized as follows. Section 2 reviews the related work. Section 3 introduces the concepts related to entity type overdependency. Section 4 presents our proposed model. Section 5 discusses the experimental performance results. Section 6 concludes the paper.

2. Related work

In this section, we review the related work on event argument extraction and contrastive learning.

2.1. Event argument extraction

For event argument extraction, feature-based methods [16–19], which rely on human-made features, have traditionally been explored to extract event arguments. Benefited from the rapid development of neural networks, representation-based methods can extract feature representations of instances more effectively. DMCNN [1] uses a conventional neural network and a dynamic multi-pooling operation to capture the most vital information about argument candidates. JRNN [8] utilizes discrete values to store the predicted event trigger words and event arguments, thereby enhancing the performance of EAE. RBPB [20] evaluates the probability of argument candidate co-occurrence and uses the evaluation results as constraints in EAE. Analogously, dbRNN [21] considers the interaction between argument candidates in the same event using a 3D tensor. JMETOR [7] introduces dependency parsing trees and a graph convolution network to jointly extract multiple events including event triggers and arguments.

With the development of pre-trained language models, the current EAE models have achieved remarkable performance. For example, DMBERT [5] uses BERT and a dynamic multi-pooling to obtain better feature representations. To tackle the problem on event arguments' participation with multiple role types, PLMETOR [22] employs multiple classifiers to predict role types separately. HMEAE [2] designs a hierarchical modular attention network to model the correlation among argument roles. TEXT2EVENT [23] treats EAE as a sequence-to-sequence generation task based on the pre-trained language model T5 [24]. BERD [3] proposes an encoder–decoder framework that utilizes predicted argument role information from other entities of the same sentence for EAE. To fill up the gap between the pre-training and fine-tuning paradigm of pre-trained language models, PAIE [25] uses pre-designed prompts of each event type and role-specific span selectors to jointly extract multiple event arguments. Besides, EAE is also treated as a question answering problem [26,27]. Regardless of which network structure is used, the existing EAE models focus mainly on exploring the modeling of inter-dependencies or intra-dependencies between event triggers and argument roles. However, they neglect the negative effects of the correlation between roles types and entity types for EAE. Therefore, in this paper, we propose a novel model to reduce the negative effects of entity type overdependency for EAE.

2.2. Contrastive learning

Contrastive learning has been widely used in computer vision [9,10] with self-supervised learning. In particular, SimCLR [9] uses a simplified contrastive learning loss to replace well-designed memory banks or architectures by generating positive samples with data augmentation. SupCon [13] extends SimCLR and generates positive samples from data augmentation and training instances with the same label. Recently, contrastive learning has also been used in natural language processing. CERT [11] uses contrastive learning to pretrain language models at the sentence level. CLINE [12] constructs adversarial and contrastive examples, and learns from both of them in contrastive learning to improve the robustness of pre-trained language models. SCL [28] improves the performance of pre-trained language models for few-shot learning tasks by combining contrastive learning loss and cross entropy loss. LCL [29] weighs positive samples and negative samples by learning a weighing network in supervised contrastive learning, thereby improving pre-trained language models for fine-grained text classification. Regarding event extraction, CLEVE [30] captures the abundant event knowledge from unsupervised data and the corresponding semantic structures by using self-supervised contrastive learning. Unlike the use of contrastive learning in a self-supervised manner for event extraction in CLEVE, we propose two novel contrastive learning methods in a supervised manner to achieve the goal of reducing entity type overdependency for EAE.

3. Entity type overdependency for EAE

Inspired by the empirical results from relation extraction [4], we experimentally investigate whether entity types provide most of the entity mention information for EAE. We follow the same setting of relation extraction [4] to replace entity mentions by their entity types in the widely used ACE 2005 English dataset. Then, we simply choose the representative model, DMBERT, and train it for EAE with the setting. Results show that DMBERT only drops 1% in F1 which has shown that entity types are the main information for entity mentions. In addition, we also observe that there exists 43 entity types in the ACE 2005 English dataset with 32 entity types participating in less than 5 role types [6]. Furthermore, when we only consider entity types with entity-role type co-occurrence frequency larger than 10, the number of entity types participating in less than 5 role types will be increased to 38.

As such, we observe that entity types provide key information for EAE and correlate strongly with role types. To study the possible effects of the correlation between entity types and role types, we first define entity type overdependency formally in this section. Thereafter, we report on an experiment conducted to study the correlation between entity types and role types by investigating the effects of entity type overdependency on EAE performance.

3.1. Definitions

Most EAE models consist of two main components: an encoder and a classifier. The encoder converts the instances into feature representations while the classifier determines the similarities between the feature representations of instances and the representation of each role type. As entity types are the main information for entity mentions and entity mentions provide the key information for encoding feature representations, feature representations depend on entity types' information. Let $G(r_k)$ be the group containing instances with role type r_k, $C(e_i, r_k)$ be the cluster containing instances with entity type e_i and role type r_k, and $\boldsymbol{h}^M_{r_k}$ and $\boldsymbol{h}^M_{e_i,r_k}$ be the representations of $G(r_k)$ and $C(e_i, r_k)$ which are encoded by an encoder of an EAE model M respectively. We define entity type dependency at the cluster-level as follows:

Definition 1 (*Entity Type Dependency*)**.** Given $C(e_i, r_k)$ with its representation $\boldsymbol{h}^M_{e_i,r_k}$ encoded by M, $C(e_i, r_k)$ is dependent on entity type e_i if there exists $C(e_i, r_l)$ and $C(e_j, r_k)$ $(i \neq j,\ k \neq l)$, such that:

$$\boldsymbol{h}^M_{e_i,r_k} \cdot \boldsymbol{h}^M_{e_i,r_l} > \boldsymbol{h}^M_{e_i,r_k} \cdot \boldsymbol{h}^M_{e_j,r_k} \tag{1}$$

In Definition 1, if the similarity between $C(e_i, r_k)$ and $C(e_i, r_l)$ is higher than the similarity between $C(e_i, r_k)$ and $C(e_j, r_k)$, then the cluster $C(e_i, r_k)$ is identified with entity type dependency. Entity type dependency shows the correlation between the entity type e_i and role type r_k such that the representation of $C(e_i, r_k)$ is more similar to the representation of $C(e_i, r_l)$ in G_{r_l} than $C(e_j, r_k)$ in G_{r_k}.

To further evaluate whether entity type dependency affects the performance of EAE models, we define semantic inconsistency of clusters. Generally, if the similarities of feature representations are inconsistent with their class labels, the corresponding classifier will tend to be error-prone [31]. Therefore, we define semantic inconsistency at the cluster-level as follows:

Definition 2 (*Semantic Inconsistency*)**.** Given $C(e_i, r_k)$ with its representation $\boldsymbol{h}^M_{e_i,r_k}$ encoded by M, the cluster $C(e_i, r_k)$ suffers from semantic inconsistency if there exists $G(r_l)$ $(k \neq l)$ such that:

$$\boldsymbol{h}^M_{e_i,r_k} \cdot \boldsymbol{h}^M_{r_l} > \boldsymbol{h}^M_{e_i,r_k} \cdot \boldsymbol{h}^M_{r_k} \tag{2}$$

where $\boldsymbol{h}^M_{r_l}$ and $\boldsymbol{h}^M_{r_k}$ are the representations of the groups $G(r_l)$ and $G(r_k)$ respectively.

In Definition 2, although the role type of instances in $C(e_i, r_k)$ is r_k rather than r_l, the similarity between the cluster $C(e_i, r_k)$ and the group $G(r_k)$ is lower than the similarity between the cluster $C(e_i, r_k)$ and the group $G(r_l)$. In other words, $C(e_i, r_k)$ should be closer to the group to which it belongs than other groups. Else, semantic inconsistency will occur and the classifier for the encoder M will tend to be error-prone. Therefore, semantic inconsistency can be used to determine whether the clusters, which are dependent on entity types, affect the performance of EAE models. Based on entity type dependency and semantic inconsistency, we define entity type overdependency at the cluster-level as follows:

Definition 3 (*Entity Type Overdependency*)**.** Given $C(e_i, r_k)$ and its representation $\boldsymbol{h}^M_{e_i,r_k}$ encoded by M, $C(e_i, r_k)$ suffers from entity type overdependency if the following conditions hold:

(a) There exists $C(e_i, r_l)$ and $C(e_j, r_k)$ $(i \neq j,\ k \neq l)$, such that $C(e_i, r_k)$ satisfies Definition 1.

(b) There exists $G(r_l)$ such that $C(e_i, r_k)$ $(k \neq l)$ satisfies Definition 2.

In Definition 3, if the cluster $C(e_i, r_k)$ satisfies the definition of entity type dependency, and the dependency further leads to semantic inconsistency that causes the classifier to be error-prone [31], then the cluster is defined as suffering from entity type overdependency. Therefore, entity type overdependency is essentially semantic inconsistency caused by entity type dependency. Take the cluster $C(e_1, r_1)$ in Fig. 1 as an example, there exists $C(e_1, r_2)$ and $C(e_5, r_1)$, such that $C(e_1, r_1)$ satisfies Definition 1. Moreover, there exists $G(r_2)$ such that $C(e_1, r_1)$ satisfies Definition 2. Thus, $C(e_1, r_1)$ suffers from entity type overdependency.

3.2. Experiment

In this section, we evaluate whether entity type overdependency affects EAE performance.

Table 2
Number of clusters and performance in F1 (%) of different types of clusters.

Model	ETD Clusters (#clusters/F1)	SI Clusters (#clusters/F1)	ETO Clusters (#clusters/F1)	Non-ETO Clusters (#clusters/F1)	All Clusters (#clusters/F1)
AttRNN	100/40.6	56/38.9	48/**18.1**	108/63.6	156/50.9
DMCNN	112/46.0	76/32.6	52/**15.6**	96/71.4	148/53.5
DMBERT	95/38.6	48/40.7	31/**15.3**	130/68.9	161/57.2

Table 3
Statistics on number of instances in "ETO Clusters".

Model	#Instances in ETO Clusters	Total #Instances	ETO Proportion
AttRNN	2845	4013	70.9
DMCNN	2799	3630	77.1
DMBERT	1281	4081	31.4

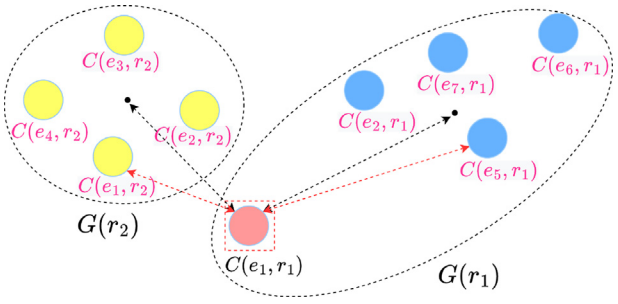


Fig. 1. The presentation of entity type overdependency for the given cluster $C(e_1, r_1)$. We use distance to express the magnitude of similarity: The closer the distance between two clusters, the higher the similarity between them.

Dataset and models. For the experiment, we choose the widely used benchmark dataset, the ACE 2005 English corpus. As RNN, CNN and BERT are the three most commonly used encoders for EAE, we train three different EAE models including AttRNN,[1] DMCNN [1] and DMBERT [5] based on the training set of the benchmark dataset for EAE, which contains 26 708 instances. Moreover, as the data distribution between the training set and the test set is similar, we use the test set to evaluate the performance of the clusters according to the definitions of entity type dependency, semantic inconsistency and entity type overdependency. As the test set of EAE relies on the detection results of event detection, we follow [2,3] and use the trained AttRNN, DMCNN and DMBERT to detect the events and the corresponding trigger words. Note that the number of instances used for testing depends on the number of correctly predicted events by the different event detection models. In this experiment, the numbers of instances in the test set are 4013/3630/4081 for AttRNN/DMCNN/DMBERT respectively.

Procedures. First, we use the encoder of each model to encode each instance from its respective test set to obtain its feature representation. Next, based on the golden role types, we divide all the instances of the test set into different groups in which the role types of the instances are the same. Thereafter, we further divide the instances in each group into different clusters according to entity types. Therefore, the entity types and role types of the instances in the same cluster are the same. As a result, for different encoders, the total numbers of clusters in the test set are 156/148/161 for AttRNN/DMCNN/DMBERT, respectively. Then, we compute the representation of each cluster/group by averaging the feature representations of the instances in that cluster/group. This simply avoids introducing additional parameters and reduces computational complexity. Then, based on the calculated representations, we classify each cluster according to the definitions of entity type dependency (ETD), semantic inconsistency (IS) and entity type overdependency (ETO). Finally, we compute the performance (in F1) of each type of clusters by treating them as test subsets.

3.3. Observations

Table 2 shows the number of clusters and performance results for different types of clusters. From Table 2, we observe that the performance of "ETO Clusters" is highly degraded regardless of which EAE model is used. The large performance difference between "ETO Clusters" and other clusters shows that the EAE models have difficulty in handling entity type overdependency. Moreover, the performance difference between "ETO Clusters" and "SI Clusters" shows that entity type overdependency, which is caused by the correlation between entity types and role types, further degrades the performance of EAE compared with semantic inconsistency. Besides, the performance difference between "Non-ETO Clusters" and "All Clusters" shows that the overall performance of each EAE model can be improved if the entity type overdependency problem is eliminated.

As the numbers of the instances in different clusters are imbalanced, we also define "ETO Proportion" at the instance-level rather than the cluster-level. Table 3 shows the "ETO Proportion" which is calculated based on the number of instances in "ETO Clusters"/"All Clusters". It indicates the degree of entity type overdependency. However, the degree of entity type overdependency and the overall performance of each model are not necessary to be a directly proportional relationship. For example, DMCNN (53.5%) outperforms AttRNN (50.9%) in F1, but the degree of entity type overdependency for DMCNN (77.1%) is higher than that of AttRNN (70.9%). It is because the encoder network structure of each EAE model is unique. In Section 5.4, we will evaluate whether the overall performance can be improved if "ETO Proportion" is reduced as implemented in our proposed model.

From the experiment, we can observe that entity type overdependency degrades the overall performance of EAE. Therefore, it is important to tackle the entity type overdependency problem encountered by the current EAE models, which mainly use RNN, CNN or BERT as their encoder. To tackle this problem, one promising direction is to reduce "ETO Proportion". As "ETO" is caused by entity type dependency for EAE, we can tackle the problem by following Definition 1 in two ways. First, we can increase the similarity between the learned feature representations of instances with the same role type but having different entity types. Second, we can also reduce the similarity between the learned feature representations of instances with different role types but having the same entity type. To achieve this, we have incorporated the above two approaches into our proposed model to tackle the entity type overdependency problem to improve the performance of the EAE task.

[1] RNN-based EAE models, such as JRNN [8] and dbRNN [21], do not release their source codes and some implementation details are not clear. Thus, we simply use a Bi-LSTM with a self-attention mechanism [32], named AttRNN, as the RNN-based baseline model and its hyperparameters are presented in Section 5.1.

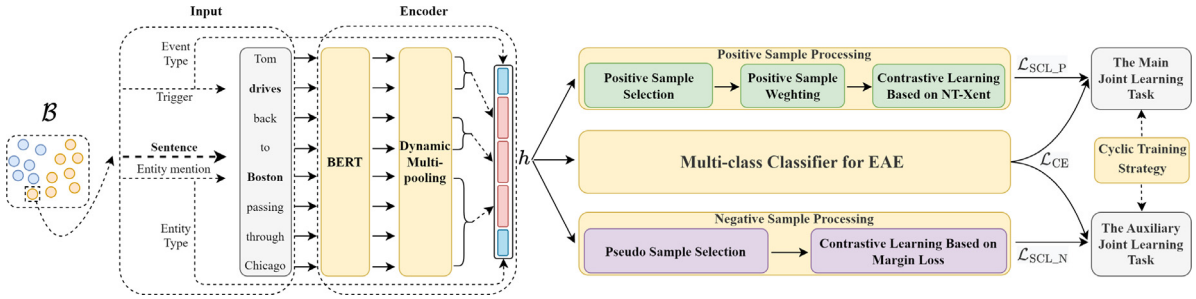


Fig. 2. Architecture of our proposed model.

4. Proposed model

In this paper, we propose a novel Multi-view Entity Type Overdependency Reduction (METOR) model to tackle the entity type overdependency problem. Fig. 2 shows the overall architecture of the proposed model which consists of four modules: Encoder, Positive Sample Processing, Negative Sample Processing and Cyclic Training Strategy.

4.1. Encoder

Encoder aims to encode each instance into a feature representation. As EAE does not involve the event detection task, we follow the previous work [2,3] by using the event detection model in [5] to predict the trigger words and their corresponding event types, and treating all entity mentions as event argument candidates before EAE. Since there may have multiple event argument candidates in a sentence, we split these candidates into multiple instances. For each event argument candidate, an instance $x = \{w_1, \ldots, tri, \ldots, arg, \ldots, w_n\}$ is created from the n-word input sentence, where arg is the argument candidate with the entity type e and tri is the predicted event trigger with the event type t. Thereafter, BERT [33] is used to encode the input sentence into hidden representations:

$$\left\{\hat{\boldsymbol{h}}_1, \ldots, \hat{\boldsymbol{h}}_{p_{tri}}, \ldots, \hat{\boldsymbol{h}}_{p_{arg}}, \ldots, \hat{\boldsymbol{h}}_n\right\} = \text{BERT}(w_1, \ldots, tri, \ldots, arg, \ldots, w_n) \tag{3}$$

where p_{tri} and p_{arg} represent the positions of tri and arg respectively.

After that, we use a dynamic multi-pooling operation to aggregate the hidden representations piecewisely as follows:

$$\left[\boldsymbol{h}_{1,p_{tri}}\right]_k = \max\left\{\left[\hat{\boldsymbol{h}}_1\right]_k \ldots, \left[\hat{\boldsymbol{h}}_{p_{tri}}\right]_k\right\} \tag{4}$$

$$\left[\boldsymbol{h}_{p_{tri}+1,p_{arg}}\right]_k = \max\left\{\left[\hat{\boldsymbol{h}}_{p_{tri}+1}\right]_k \ldots, \left[\hat{\boldsymbol{h}}_{p_{arg}}\right]_k\right\} \tag{5}$$

$$\left[\boldsymbol{h}_{p_{arg}+1,n}\right]_k = \max\left\{\left[\hat{\boldsymbol{h}}_{p_{arg}+1}\right]_k \ldots, \left[\hat{\boldsymbol{h}}_n\right]_k\right\} \tag{6}$$

where $[\cdot]_k$ represents the kth value of a vector. Then, we randomly initialize all event type labels and all entity type labels into embedding matrices, which are denoted as $\boldsymbol{W}_T \in \mathbb{R}^{n_t \times d_s}$ and $\boldsymbol{W}_E \in \mathbb{R}^{n_e \times d_s}$ respectively. Note that d_s, n_t and n_e are the dimension of event type/entity type embeddings, the total number of event types and the total number of entity types of the given EAE dataset, respectively. Further, we use the event type t to look up the embedding matrix $\boldsymbol{W}_T$, denoted as $\boldsymbol{W}_T(t)$. Similarity, $\boldsymbol{W}_E(e)$ is obtained for the entity type e. Finally, we concatenate the above representations as the feature representation $\boldsymbol{h} \in \mathbb{R}^{d_{out}}$ of the instance:

$$\boldsymbol{h} = \left[\boldsymbol{h}_{1,p_{tri}}; \boldsymbol{h}_{p_{tri}+1,p_{arg}}; \boldsymbol{h}_{p_{arg}+1,n}; \boldsymbol{W}_T(t); \boldsymbol{W}_E(e)\right] \tag{7}$$

4.2. Positive sample processing

In this module, we propose a select-then-weigh contrastive learning method which consists of three steps: positive sample selection, positive sample weighing and contrastive learning based on Normalized Temperature-scaled Cross Entropy (NT-Xent). First, we select positive samples based on the entity type for a given instance, which enables the EAE model to focus on processing the positive samples with entity types apart from the given instance. Next, global co-occurrence information and semantic relevance information are computed for weighing the relevance of the selected positive samples. As such, the EAE model can increase the contribution of the positive samples which hold the key semantic information for the role type of the given instance, and learn more effective semantic information besides entity types. Finally, we obtain the loss of the select-then-weigh contrastive learning method based on NT-Xent.

Positive sample selection. Given a batch $\mathcal{B}$ containing K instances, for each instance x_i in it, the instances whose role type is the same as x_i are first selected to generate the positive set $\mathcal{P}(x_i)$, while the negative set $\mathcal{N}(x_i)$ is generated if otherwise. Based on $\mathcal{P}(x_i)$, we perform a simple selection strategy to generate a new positive set as follows:

$$\mathcal{P}_s(x_i) = \left\{x_p : x_p \in \mathcal{B}, (e_p \neq e_i) \wedge (r_p = r_i) \wedge (p \neq i)\right\} \tag{8}$$

where r_i and e_i denote the role type and entity type of the instance x_i respectively. Note that we only apply the selection if $\mathcal{P}(x_i)$ contains more than one sample to avoid the data sparsity problem.

Positive sample weighing. After the selection, we assign a weight to each positive sample in $\mathcal{P}_s(x_i)$ according to its importance on predicting r_i. The vanilla contrastive learning method [13] treats all positive samples equally in the contrastive learning loss. Therefore, it is not appropriate for this work as the positive samples with different entity types should be distinguished. Thus, we need to calculate the importance of different positive samples for x_i based on its role type r_i, and increase the weight of those instances whose entity types are more important for predicting r_i, thereby forcing the model to learn from instances which hold more relevant semantic information for r_i. To achieve this, we compute the importance of different positive samples for x_i based on its role type r_i according to the global co-occurrence information and semantic relevance information.

The global co-occurrence information records the number of co-occurrences for different entity types and role types from the entire training set into a co-occurrence matrix $I_{e,r}$. Let $\mathcal{E}$ and $\mathcal{R}$ be the set of entity types and role types in the training set, respectively. Assume that the pth instance (i.e., x_p) in the batch $\mathcal{B}$ is in $\mathcal{P}_s(x_i)$ and its entity type is e_p. Inspired by TF-IDF [34], we define Entity Type Frequency (ETF) as the frequency of e_p

occurring with r_i as follows:

$$\mathrm{ETF}\left(e_p, r_i\right) = \frac{I_{e_p,r_i}}{\sum_{m\in\mathcal{E}} I_{m,r_i}} \tag{9}$$

Thereafter, we use the number of role types which occur with e_p to compute the importance of e_p. The smaller the number of role types is, the more important e_p is. Thus, we define Inverse Role Type Frequency (IRTF) as follows:

$$\mathrm{IRTF}\left(e_p\right) = \log \frac{|\mathcal{R}|}{|\{r \in \mathcal{R} : I_{e_p,r} > 0\}| + 1} \tag{10}$$

We then obtain the importance score $w_1\left(x_p, x_i\right)$ of x_p to x_i by considering ETF and IRTF based on the global co-occurrence information as follows:

$$w_1\left(x_p, x_i\right) = \mathrm{ETF}\left(e_p, r_i\right) \times \mathrm{IRTF}\left(e_p\right) \tag{11}$$

Entity mentions with the same entity type tend to have different role types in instances with different event types. Thus, for different event types, even the entity types of the instances are the same, the importance to the given instance could be different. However, using the global co-occurrence information alone is not sufficient to capture the influence of different event types when considering the importance of x_p to x_i. Therefore, we also utilize the semantic information learned by the model. To do this, we first randomly initialize all role type labels into embedding matrices, denoted as $\boldsymbol{W}_R \in \mathbb{R}^{n_r \times d_s}$, where n_r is the total number of role types (including the special role type "None") and d_s is the dimension of role type embeddings. Next, for the given instance x_i, we obtain the event-aware entity type and role type representation:

$$\boldsymbol{e}_i^t = \boldsymbol{W}_T\left(t_i\right) \odot \boldsymbol{W}_E\left(e_i\right) \tag{12}$$

$$\boldsymbol{r}_i^t = \boldsymbol{W}_T\left(t_i\right) \odot \boldsymbol{W}_R\left(r_i\right) \tag{13}$$

where $\odot$ denotes the element-wise multiplication, and r_i, e_i and t_i denote the role type, entity type and event type of the instance x_i respectively. $\boldsymbol{W}_T\left(t_i\right)$ denotes the embedding by looking up the embedding matrix $\boldsymbol{W}_T$ with the event type t_i. $\boldsymbol{W}_E\left(e_i\right)$ and $\boldsymbol{W}_R\left(r_i\right)$ are similarly obtained. Note that $\boldsymbol{W}_T$ and $\boldsymbol{W}_E$ are also used as part of the input in Eq. (7). Therefore, the learned semantic information of event types and entity types is shared between different modules. Similarly, we can also obtain $\boldsymbol{e}_p^t$ for any positive sample x_p. Then, we use the additive attention to calculate the importance score of x_p to x_i based on the semantic relevance information and employ the softmax function to obtain the normalized importance score $w_2\left(x_p, x_i\right)$:

$$s\left(x_p, x_i\right) = \boldsymbol{v}^\top \tanh\left(\boldsymbol{U}\left[\boldsymbol{e}_i^t; \boldsymbol{e}_p^t; \boldsymbol{r}_i^t\right]\right) \tag{14}$$

$$w_2\left(x_p, x_i\right) = \frac{\exp\left(s\left(x_p, x_i\right)\right)}{\sum_{e_j\in\mathcal{E}} \exp\left(s\left(x_j, x_i\right)\right)} \tag{15}$$

where $\boldsymbol{v} \in \mathbb{R}^{3d_s}$ and $\boldsymbol{U} \in \mathbb{R}^{3d_s \times 3d_s}$ are trainable vector and matrix respectively. After that, we obtain the final importance score of x_p to x_i as follows:

$$\mathrm{w}\left(x_p, x_i\right) = \alpha w_1\left(x_p, x_i\right) + (1-\alpha) w_2\left(x_p, x_i\right) \tag{16}$$

where $\alpha\,(0 < \alpha < 1)$ is a weight parameter.

Contrastive learning based on NT-Xent. Following [9,13], the feature representation $\boldsymbol{h}_i$ of instance x_i is first mapped into a new space, where contrastive learning is used to improve the quality of learning, as follows:

$$\boldsymbol{z}_i = \boldsymbol{W}^2 \sigma\left(\boldsymbol{W}^1 \boldsymbol{h}_i\right) \tag{17}$$

where $\boldsymbol{W}^1 \in \mathbb{R}^{d_1 \times d_{out}}$ and $\boldsymbol{W}^2 \in \mathbb{R}^{d_2 \times d_1}$ are trainable matrices, and σ is a ReLU activation function. Note that d_1 and d_2 are dimensional parameters. Through positive sample selection and weighing, we can obtain the contrastive learning loss $\mathcal{L}_{\mathrm{SCL\_P}}$ based on NT-Xent [9] as follows:

$$\mathcal{L}_{\mathrm{SCL\_P}} = \sum_{i=1}^{K} \frac{-1}{|\mathcal{P}_s\left(x_i\right)|} \sum_{p\in\mathcal{P}_s(x_i)} \mathrm{w}(x_p, x_i) \cdot \log \frac{\exp\left(\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right)/\tau\right)}{\sum_{n\in K\setminus i} \exp\left(\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_n\right)/\tau\right)} \tag{18}$$

where $\mathrm{sim}\left(\boldsymbol{z}_i, \boldsymbol{z}_p\right) = \boldsymbol{z}_i^\top \boldsymbol{z}_p / \|\boldsymbol{z}_i\| \, \|\boldsymbol{z}_p\|$ and τ is a temperature parameter.

4.3. Negative sample processing

In this module, we propose a pseudo-positive contrastive learning method that comprises two steps: pseudo sample selection and contrastive learning based on margin loss. First, we select pseudo samples from negative samples based on entity types. Then, we use these selected samples as the contrastive reference to reduce the similarity of the learned feature representations between the given instance and the negative samples whose entity types are the same as the given instance.

Pseudo sample selection. Given a batch $\mathcal{B}$ containing K instances, for each instance x_i in it, we first select pseudo-positive samples and pseudo-negative samples. The sampling selection process is performed as follows:

$$\mathcal{P}_{pseudo}\left(x_i\right) = \left\{x_p : x_p \in \mathcal{B}, \left(e_p \neq e_i\right) \wedge \left(r_p \neq r_i\right) \wedge \left(p \neq i\right)\right\} \tag{19}$$

$$\mathcal{N}_{pseudo}\left(x_i\right) = \left\{x_n : x_n \in \mathcal{B}, \left(e_n = e_i\right) \wedge \left(r_n \neq r_i\right) \wedge \left(n \neq i\right)\right\} \tag{20}$$

where the pseudo-positive sample set and pseudo-negative sample set for instance x_i are denoted as $\mathcal{P}_{pseudo}\left(x_i\right)$ and $\mathcal{N}_{pseudo}\left(x_i\right)$ respectively.

Contrastive learning based on margin loss. First, we average the feature representations of the instances in $\mathcal{P}_{pseudo}\left(x_i\right)$ and $\mathcal{N}_{pseudo}\left(x_i\right)$, denoted as $\boldsymbol{z}_i^P$ and $\boldsymbol{z}_i^N$ respectively, to represent the overall information of these two sets. Then, we employ $\mathcal{P}_{pseudo}\left(x_i\right)$ as the contrastive reference to conduct contrastive learning to reduce the similarity between the instance x_i and the negative samples that belong to $\mathcal{N}_{pseudo}\left(x_i\right)$. As these pseudo samples are negative samples, it is not suitable to use the instance-level contrastive learning loss NT-Xent to increase the similarity between pseudo-positive samples and the given instance. Therefore, we use a margin loss to contrast the overall representation of the pseudo-positive sample set and the pseudo-negative sample set as follows:

$$\mathcal{L}_{\mathrm{SCL\_N}} = \sum_{i=1}^{K} \max\left\{\boldsymbol{z}_i^N \cdot \boldsymbol{z}_i - \boldsymbol{z}_i^P \cdot \boldsymbol{z}_i + \gamma, 0\right\} \tag{21}$$

where γ is a margin hyperparameter.

4.4. Cyclic training strategy

In this module, we first obtain the EAE loss. Given the instance x, we feed its feature representation h, which is obtained from the Encoder, into a multi-class classifier to calculate the probability distribution $p\left(x\right)$ as follows:

$$p\left(x\right) = \mathrm{softmax}\left(\boldsymbol{W}^c \boldsymbol{h} + \boldsymbol{b}^c\right) \tag{22}$$

where $\boldsymbol{W}^c \in \mathbb{R}^{n_r \times d_{out}}$ and $\boldsymbol{b}^c \in \mathbb{R}^{n_r}$ are the parameters of the classifier to be trained. Given a batch $\mathcal{B}$ containing K instances,

the cross entropy loss $\mathcal{L}_{\text{CE}}$ is used to obtain the EAE loss as follows:

$$\mathcal{L}_{\text{CE}} = -\sum^{K} \log p\,(r|x) \tag{23}$$

where $p\,(r|x)$ denotes the estimated probability of the golden role type r for the instance x. During testing, $p\,(x)$ is used to obtain the predicted role type for EAE.

After that, we note that optimizing $\mathcal{L}_{\text{SCL\_N}}$ inevitably leads to the increasing of similarity between the instance x_i and the negative samples that belong to $\mathcal{P}_{pseudo}\,(x_i)$. However, optimizing $\mathcal{L}_{\text{SCL\_P}}$ leads to a reduction in the similarity between the instance x_i and all negative samples according to Eq. (18). Thus, $\mathcal{L}_{\text{SCL\_N}}$ is inconsistent with the objective of $\mathcal{L}_{\text{SCL\_P}}$ and we should not simply jointly optimize them. Therefore, we propose a cyclic training strategy that uses two different joint training goals in turn to reconcile the objectives of $\mathcal{L}_{\text{SCL\_P}}$ and $\mathcal{L}_{\text{SCL\_N}}$. In particular, the joint optimization of $\mathcal{L}_{\text{SCL\_P}}$ and $\mathcal{L}_{\text{CE}}$ is considered as the main joint learning task which is first trained for f epochs, where f is a hyperparameter. Then, the joint optimization of $\mathcal{L}_{\text{SCL\_N}}$ and $\mathcal{L}_{\text{CE}}$ is considered as the auxiliary joint learning task which is trained for one epoch. This process is repeated until the training is completed. As a result, the inconsistency between the training objectives of $\mathcal{L}_{\text{SCL\_P}}$ and $\mathcal{L}_{\text{SCL\_N}}$ can be gradually adjusted and successfully dealt with. The proposed training strategy is given as follows:

$$\mathcal{L}_{\text{JOINT}} = \begin{cases} \mathcal{L}_{\text{SCL\_P}} + \mathcal{L}_{\text{CE}} & epoch \bmod f \neq 0 \\ \mathcal{L}_{\text{SCL\_N}} + \mathcal{L}_{\text{CE}} & epoch \bmod f = 0 \end{cases} \tag{24}$$

where *epoch* denotes the number of trained epochs.

5. Performance evaluation

Apart from using BERT in the Encoder, we also use the encoders of AttRNN and DMCNN [1] to obtain feature representations for our model, which are named as METOR (RNN) and METOR (CNN) respectively. As EAE relies on the detection results of the event detection task, we follow [2,3] and use the trained AttRNN, DMCNN and DMBERT to detect the events and the corresponding trigger words for METOR (RNN), METOR (CNN) and METOR respectively. In this section, we present the experimental setup, performance results, ablation study, further analysis on entity type information, a case study, hyperparameter sensitivity analysis and computational complexity analysis.

5.1. Experimental setup

In this section, we discuss the dataset, metrics, hyperparameters and the baseline models.

Dataset. For the last five years, almost all top publications for EAE, including the latest state-of-the-art model BERD [3], used the ACE 2005 English dataset only in their performance evaluation. Wang et al. [2] also used the TAC KBP 2016 dataset. However, the dataset is not available for open access. Thus, for fair comparison with the latest models, we follow most EAE works [3,7,8,22,26] to conduct the experiments based on the ACE 2005 English corpus. The dataset contains 599 documents, 33 event types, 46 entity types and 35 argument role types. Note that we use "None" as a special role type to represent the corresponding argument candidate which plays no role in a given instance for the EAE task. Same as previous works [1,3,35,36], the dataset is split into 529, 30 and 40 documents as training, development and test sets respectively.

Metrics. We follow the standard criteria of the EAE task. If the event type, offsets and argument role are the same as the golden annotation, then the argument candidate is correctly classified. The offsets refer to the start and end positions of the argument candidate which have already been given in the dataset. The micro-averaged precision (P), recall (R) and F1 score (F1) are used as the evaluation metrics.

Hyperparameters. In the Encoder, the 100-dimensional pre-trained Glove word embeddings, 5-dimensional randomly initialized event type embeddings, 5-dimensional randomly initialized entity type embeddings and 5-dimensional randomly initialized position embeddings are included in the input embeddings for METOR (RNN). Then, Bi-LSTM and the biaffine attention mechanism [37] are used subsequently. As to METOR (CNN), we use the same input embeddings and hyperparameters as DMCNN [1]. Similarly, we also add the 5-dimensional randomly initialized entity type embeddings to the input embeddings for METOR (CNN). For METOR, we use $\text{BERT}_{\text{BASE}}$ to encode sentences and the entity type embedding dimension d_s is set to 50. And d_{out} is 900/1500/2404 for METOR (RNN), METOR (CNN) and METOR respectively. In Positive Sample Processing, we set d_1 to 512, d_2 to 512, the temperature parameter τ to 0.1 and the weight α to 0.7. In Negative Sample Processing, we set γ to 0.1. In Cyclic Training Strategy, the cyclic frequency f is set to 4, 3 and 3 for METOR (RNN), METOR (CNN) and METOR respectively. Moreover, we set the batch size as 80 and epoch as 10 on a NVIDIA Tesla V100 GPU for different encoders during training. AdamW is used as the optimizer for METOR (RNN), METOR (CNN) and METOR with learning rate of 1×10^{-3}, 1×10^{-3} and 5×10^{-5} respectively.

Baselines. We compare our proposed model with the following models:

- DMCNN [1] - It proposes a CNN and a dynamic multi-pooling operation to extract feature representations.
- JRNN [8] - It proposes a RNN with discrete memory matrices that utilizes inter-dependencies between event triggers and argument roles.
- dbRNN [21] - It proposes dependency bridges over RNN with a tensor layer that utilizes syntactical information and argument-argument interactions.
- DMBERT [5] - It proposes BERT and a dynamic multi-pooling operation to extract feature representations.
- PLMETOR [22] - It proposes a BERT-based model to predict argument roles and generate extra labeled instances to further improve the performance.
- HMEAE [2] - It proposes a hierarchical modular attention network that utilizes the correlations of argument roles.
- BERT (Inter) [3] - It proposes a BERT-based model with inter-dependencies [8].
- BERT (Intra) [3] - It proposes a BERT-based model with argument-argument interactions [21].
- BERD [3] - It proposes an encoder–decoder framework that utilizes intra-event argument interactions.

Moreover, some recent works follow DyGIE++ [38] by keeping only 22 argument role types,[2] and use the golden triggers and corresponding event types as the input. To avoid confusion, we denote the ACE 2005 English dataset under the above experimental setting as ACE05-R+GT. We train our model based on ACE05-R+GT and compare the performance with the following models:

[2] Most of the omitted role types are time-related types, such as "*Time-Within*" and "*Time-Starting*".

Table 4
Experimental results based on the ACE 2005 English dataset.

Model	P(%)	R(%)	F1(%)
AttRNN	50.6	51.1	50.9
DMCNN	62.2	46.9	53.5
JRNN	54.2	56.7	55.4
dbRNN	**66.2**	52.8	58.7
HMEAE (CNN)	57.3	54.2	55.7
METOR (RNN)	51.9	58.0	54.8
METOR (CNN)	56.2	61.1	58.5
+ BERT (*base*)			
DMBERT	58.8	55.8	57.2
PLMETOR	62.3	54.2	58.0
BERT (Inter)	58.4	57.1	57.8
BERT (Intra)	56.4	61.2	58.7
HMEAE (BERT)	62.2	56.6	59.3
BERD	59.1	61.5	60.3
METOR	60.3	**64.7**	**62.4**†

† denotes that our model significantly outperforms the best baseline BERD with $p < 0.01$ under a paired two-sided t-test.

Table 5
Experimental results based on ACE05-R+GT. PLM is the pre-trained language model used by the corresponding EAE model and Version denotes the version of the PLM.

Model	PLM	Version	F1(%)
EEQA	BERT	*base*	65.4
EEQA-BART	BART		67.7
BART-Gen	BART		55.0
PAIE	BART		69.8
METOR	BERT		70.6
EEQA	BERT	*large*	68.9
EEQA-BART	BART		72.2
BART-Gen	BART		66.7
DEGREE	BART		73.5
PAIE	BART		72.7
METOR	BERT		**74.6**‡

‡ denotes that our METOR model significantly outperforms the best baseline DEGREE with $p < 0.05$ under a paired two-sided t-test.

- EEQA [26] - It treats event extraction including EAE as a question answering problem with rule-based question generation strategies.
- EEQA-BART [25] - It is based on EEQA [26] with BART as the pre-trained language model for achieving better performance.
- BART-Gen [39] - It treats EAE as a conditional generation task based on a given context and an unfilled template.
- DEGREE [40] - It utilizes label semantics and shares knowledge between ED and EAE in an end-to-end generation-based event extraction framework.
- PAIE [25] - It designs multi-role prompts to obtain role-specific representations which are then used in a jointly optimal question answering framework.

Note that EEQA-BART, BART-Gen, DEGREE and PAIE use the pre-trained model BART [41], which is based on Transformer [42] and widely used in generation-based models.

5.2. Performance results

Table 4 shows the experimental results on the ACE 2005 English dataset. From the results, we observe that METOR (RNN) achieves 3.9% improvement on F1 compared with AttRNN with the same RNN encoder for EAE. Moreover, METOR (RNN) achieves comparative performance with JRNN that utilizes the interactions between event triggers and arguments for EAE. However, dbRNN outperforms METOR (RNN). We attribute it to two reasons. First, dbRNN utilizes extra syntactical information and argument-argument interactions while METOR (RNN) simply predicts each argument candidate individually. Second, dbRNN optimizes the performance of ED and EAE tasks simultaneously while METOR (RNN) does not benefit from such joint optimization. When we use CNN as the encoder, METOR (CNN) achieves 5% improvement on F1 when compared with DMCNN. Also, we can observe that METOR (CNN) achieves 2.8% improvement on F1 over HMEAE (CNN), which uses the encoder of DMCNN to obtain feature representations. Between the models that do not utilize any pre-trained language models, METOR (CNN) outperforms most EAE models and achieves competitive performance when compared with dbRNN. In addition, we also observe that METOR (CNN) outperforms DMBERT, PLMETOR and BERT (Inter) which use BERT as the encoder.

When we use BERT as the encoder for EAE, METOR achieves the state-of-the-art performance[3] (in F1) and outperforms the latest state-of-the-art model BERD by 2.1% in F1. Moreover, METOR outperforms DMBERT by 5.2% in F1. Overall, as our proposed METOR model can learn more semantic information besides entity types to reduce the entity type overdependency problem as discussed in Section 3, it has outperformed the different EAE baseline models and achieved the state-of-the-art performance for EAE.

As for ACE05-R+GT, the experimental results are shown in Table 5. From the results, we observe that METOR achieves the state-of-the-art performance in F1 when BERT (*large*) is used. Moreover, if our METOR model uses BERT (*base*) in the Encoder, it will outperform all the other models which use the base-version of PLMs, and achieves competitive performance when compared with those models which use the large-version of PLMs. Also, we can observe that EEQA-BART outperforms EEQA regardless of which version of PLMs is used. It shows that better performance can be obtained in EEQA with BART as PLM than BERT. For future work, we will explore using BART in our model to further improve the performance.

Table 6
Ablation studies based on the ACE 2005 English dataset for our model.

Model	P(%)	R(%)	F1(%)	Δ F1
METOR (RNN)	51.9	58.0	54.8	–
w/o Positive Sample Processing	48.7	55.7	52.0	−2.8
w/o Negative Sample Processing	51.1	57.2	54.0	−0.8
w/o Cyclic Training Strategy	50.8	52.7	51.7	−3.1
METOR (CNN)	56.2	61.1	58.5	–
w/o Positive Sample Processing	52.9	59.2	55.9	−2.6
w/o Negative Sample Processing	57.4	56.3	56.8	−1.7
w/o Cyclic Training Strategy	54.0	57.4	55.7	−2.8
METOR	60.3	64.7	62.4	–
w/o Positive Sample Processing	59.7	61.2	60.4	−2.0
w/o Negative Sample Processing	59.3	62.9	61.1	−1.3
w/o Cyclic Training Strategy	57.1	62.1	59.5	−2.9

Δ F1 is the performance difference in F1 when compared with the full model.

5.3. Ablation study

We conduct an ablation study to analyze the effectiveness of each module in our model. As shown in Table 6, all the three modules contribute significantly to the performance of our proposed model. Here, we focus the discussion on METOR. Firstly, if we remove the Positive Sample Processing and Negative Sample

[3] We note that RCEE_ER [27] proposed in 2020 has achieved 63.6% in F1 score. However, RCEE_ER has benefited from extra resources including data argumentation and unsupervised data. Thus, we follow BERD [3] which was proposed in 2021 and do not compare our proposed model with RCEE_ER owing to the difference in the use of external resources.

Table 7
Performance comparison between our model with the two kinds of variant models which use the entity type information.

Model	P(%)	R(%)	F1(%)	Δ F1
AttRNN	50.6	51.1	50.9	–
AttRNN+type	51.8	51.6	51.7	+0.8
AttRNN+type+CL	50.2	53.4	51.7	+0.8
METOR (RNN)	51.9	58.0	54.8	+3.9
DMCNN	62.2	46.9	53.5	–
DMCNN+type	51.5	58.3	54.7	+1.2
DMCNN+type+CL	53.8	56.1	54.9	+1.4
METOR (CNN)	56.2	61.1	58.5	+5.0
DMBERT	58.8	55.8	57.2	–
DMBERT+type	55.7	61.3	58.4	+1.2
DMBERT+type+CL	55.9	62.4	59.0	+1.8
METOR	60.3	64.7	62.4	+5.2

Δ F1 is the performance difference in F1 when compared with the corresponding baseline EAE models.

Processing modules, the performance of the proposed model will be dropped by 2.0% and 1.3% respectively in F1. It demonstrates that reducing entity type overdependency from positive samples and negative samples can both effectively help improve the performance of the proposed model. Secondly, if we replace Cyclic Training Strategy by summing the losses of different modules, the performance of the proposed model will be decreased by 2.9% in F1. This shows that Cyclic Training Strategy can reconcile the different contrastive learning objectives from Positive Sample Processing and Negative Sample Processing effectively. Thirdly, we observe that our proposed METOR model outperforms the state-of-the-art model BERD even if either the Positive Sample Processing module or Negative Sample Processing module is removed. This is because these two modules aim to push the model to learn more semantic information besides entity type information to tackle the entity type overdependency problem.

5.4. Further analysis on entity type information

As an entity-based task, there are both advantages and disadvantages of entity type information for the EAE task. On one hand, entity type information provides the main source of information on entity mentions for EAE. On the other hand, entity type information may cause entity type overdependency which may degrade the overall performance of EAE. Therefore, it is important to tackle the entity type overdependency problem to improve the EAE performance. In this section, we present an analysis on the effectiveness of entity type information and contrastive learning, and the effects of entity type overdependency on the performance of our proposed METOR model.

Analysis on entity type information and contrastive learning. In the experiment, we use two kinds of variant models, namely "Baseline+type" and "Baseline+type+CL", where "Baseline" can be AttRNN, DMCNN or DMBERT for our analysis. Note that these "Baseline" models do not use entity type information. For "Baseline+type", we use entity type information as part of the input. Moreover, for "Baseline+type+CL", in addition to using entity type information as input, we add the loss from the vanilla supervised contrastive learning method [13] to the cross-entropy loss in Eq. (23) from Cyclic Training Strategy.

Table 7 shows the performance comparison between our model with the two kinds of variant models. From the results, we can observe that "Baseline+type" can improve the performance of AttRNN, DMCNN and DMBERT by 0.8%, 1.2% and 1.2% respectively in F1. It shows that using entity type information in the input of these baseline methods for EAE has achieved some improvements. Moreover, "Baseline+type+CL" can further improve the performance of "DMCNN+type" and "DMBERT+type" by 0.2% and 0.6% in F1 respectively except "AttRNN+type". It indicates that the vanilla contrastive learning method further improves the effectiveness of using entity type information for "Baseline+type". As our METOR model uses the entity type information and the proposed contrastive learning methods, it helps learn more effective semantic information from text besides entity types. Therefore, our METOR model has achieved 3.9%, 5.0%, 5.2% performance improvement in F1 when compared with AttRNN, DMCNN and DMBERT respectively.

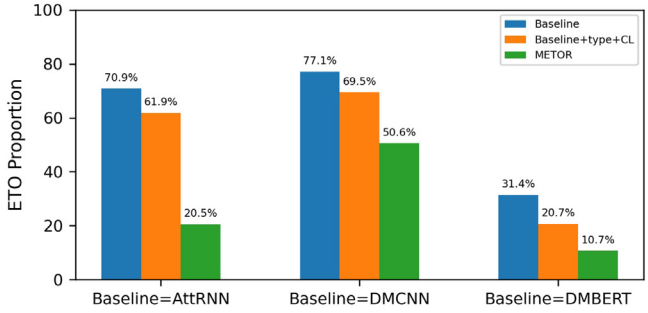


Fig. 3. Degree of entity type overdependency suffered by different methods when using different kinds of EAE encoders. "Baseline" can be AttRNN, DMCNN or DMBERT.

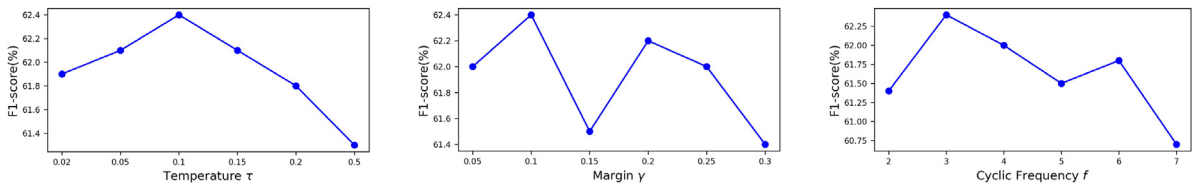
Analysis on entity type overdependency. In Section 3, we have observed that different EAE encoders suffer from entity type overdependency. The overall performance of each EAE encoder can be improved if "ETO Proportion", which indicates the degree of entity type overdependency, is reduced. To analyze the effectiveness of our METOR model in tackling entity type overdependency, we compare our METOR model with the corresponding "Baseline" and "Baseline+type+CL" models based on "ETO Proportion". As shown in Fig. 3, the "ETO Proportion" of "Baseline+type+CL" has slightly reduced when compared with the corresponding "Baseline" model showing that vanilla supervised contrastive learning can alleviate entity type overdependency to a certain extent. Moreover, we can also observe that the "ETO Proportion" of our METOR model has dropped quite significantly when compared with the corresponding "Baseline" model regardless of which encoder is used. As such, our METOR model can reduce the degree of entity type overdependency effectively. In addition, we also observe that the effectiveness of "Baseline+type+CL" in tackling entity type overdependency is not as good as our METOR model. It is because the vanilla supervised contrastive learning mainly solves the semantic inconsistency given in Definition 2.

5.5. Hyperparameter sensitivity analysis

In this section, we report on the effects of different hyperparameters on the performance of the proposed METOR model. The model hyperparameters include the temperature τ in Eq. (18), the margin γ in Eq. (21) and the cyclic frequency f in Eq. (24). These hyperparameters are critical to our proposed contrastive learning methods and cyclic training strategy.

As shown in Fig. 4(a), METOR achieves the best performance at a temperature τ of 0.1. The changes in METOR performance are within 1.1 for τ ranging from 0.02 to 0.5, which demonstrates the robustness of our proposed METOR model against different settings of τ.

From Fig. 4(b), we observe that the performance of METOR achieves the best performance when the margin γ is set to 0.1. When the margin γ is set to 0.05, the reduction in entity type



(a) (b) (c)

Fig. 4. Effects of different hyperparameters on the performance of METOR.

Table 8
A case study on six instances from the test set of ACE 2005 English dataset. The trigger words and event argument candidates are in red and blue, respectively.

Instance	Entity Type	DMBERT	DMBERT (CL)	METOR
(1) Liana Owen drove 10 h from Pennsylvania to attend the rally in Manhattan with her parents.	County-or-District	Place ✓	Place ✓	Place ✓
(2) He asked the court to invalidate the verdict and throw out the criminal case against Pasko.	Government	Adjudicator ✓	Adjudicator ✓	Adjudicator ✓
(3) Separately, former WorldCom CEO Bernard Ebbers failed on April 29 to make a first repayment of 25 million dollars on a 400-million-dollar loan from MCI, the Journal said, citing SEC documents.	Individual	Giver ×	Recipient ✓	Recipient ✓
(4) U.S. Defense Secretary Donald H. Rumsfeld discussed the resolution with Prime Minister Tony Blair and Defense Secretary Geoff Hoon on Friday as Rumsfeld returned from a tour of Iraq, Afghanistan and the Persian Gulf region.	Nation	Destination ×	Origin ✓	Origin ✓
(5) It was not clear how many people were in the cafe at the time of the blast.	Indeterminate	None ×	None ×	Target ✓
(6) The space agency has been under intense scrutiny since February when the space shuttle Columbia disintegrated over Texas, killing all seven crew members.	Air	Instrument ×	Instrument ×	None ✓

overdependency from the view of negative samples is insufficient and the performance of METOR drops slightly. When the margin γ is larger than 0.1, the performance is decreased. This is due to the excessive similarity of the learned feature representations between the given instance and the negative samples in the pseudo-positive sample set. As a result, it will cause training inconsistency between the Positive Sample Processing and Negative Sample Processing modules. However, the decrease in performance is quite limited as our proposed cyclic training strategy has greatly alleviated the training inconsistency caused by the larger margin.

From Fig. 4(c), we observe that METOR achieves the best performance when the cyclic frequency f is set to 3. When the cyclic frequency f is set to 2, there will be no difference between the training frequency of the main task and the auxiliary task in the cyclic training strategy. Thus, METOR will switch frequently between the two inconsistent training objectives, which will then degrade the performance accordingly. Moreover, we also observe that a cyclic frequency f larger than 3 leads to a lower training frequency of the Negative Sample Processing module in the cyclic training strategy. As such, METOR will be degenerated into METOR without Negative Sample Processing.

5.6. Computational complexity analysis

Although we use different encoders in our model, the analysis on computational complexity is similar. Thus, we focus on discussing the analysis on using BERT in the Encoder. The computational complexity of our model is calculated as follows: Firstly, in the Encoder module, the computational complexity of BERT mainly comes from the multi-head attention [42] with $O(N^2d + Nd^2)$, where N is the maximal length of the input sequence and d is the dimension of BERT's hidden representations. Secondly, in the Positive Sample Processing module, the computational complexity of semantic relevance information is $O((d_s^2 + d_s)n_e)$, where d_s is the dimension of event type/entity type/role type embeddings and n_e is the total number of entity types in the given EAE dataset. As for the contrastive learning based on NT-Xent, the computational complexity is $O((d + d_s + d_2)d_1 + Kd_2 + K^2)$, where d_1 and d_2 are the dimensional parameters, and K is the batch size. Thirdly, in the Negative Sample Processing module, the computational complexity is $O(d_2)$. Lastly, in the Cyclic Training Strategy module, the computational complexity of the multi-class classifier is $O(n_r(d + d_s))$, where n_r is the total number of role types in the given EAE dataset.

Therefore, the overall computational complexity of our METOR model is $O(Nd^2 + (N^2 + d_1 + n_r)d + (d_1 + K)d_2 + n_e d_s^2 + (d_1 + n_r + n_e)d_s + K^2)$. If the small parameters n_e, n_r and d_s are omitted, the computational complexity can be simplified as $O(Nd^2 + (N^2 + d_1)d + (d_1 + K)d_2 + K^2)$. Moreover, compared with DMBERT, whose computational complexity is $O(Nd^2 + N^2d)$, our model has an increase of $O(d_1d + (d_1 + K)d_2 + K^2)$. Note that $d_1 + K < d$, $d_2 < d$ and $K < N$ are set in our hyperparameter settings. Therefore, the increased complexity $O(d_1d + (d_1 + K)d_2 + K^2)$ is estimated to be close to $O(d^2 + N^2)$. Therefore, our proposed METOR model has a slightly higher computational complexity than DMBERT, but with 5.2% performance improvement in F1.

5.7. Case study

In this section, we discuss six instances from the test set of the ACE 2005 English dataset for a case study. Table 8 shows the predictions of these instances by using DMBERT, DMBERT (CL) and METOR. Note that DMBERT (CL) denotes the variant model DMBERT+type+CL shown in Table 7.

In Instance (1) where "*rally*" triggers a "*Demonstrate*" event, the entity type and role type of the event argument candidate "*Manhattan*" are "*County-or-District*" and "*Place*" respectively. As the entity type "*County-or-District*" frequently participates in the role type "*Place*" and provides the crucial information for the role

type prediction, an event argument candidate is likely to play the "*Place*" role in the event if it represents a "*County-or-District*" object. Thus, DMBERT, DMBERT (CL) and METOR give the correct role type prediction with the entity type "*County-or-District*" that provides the semantic information.

In Instance (2), the entity type and role type of the event argument candidate "*the court*" are "*Government*" and "*Adjudicator*" respectively. Although the entity type "*Government*" cannot provide any direct semantic information for predicting the role type, a useful cue can be obtained when the "*Convict*" event triggered by "*verdict*" is given. In other words, the "*Government*" object is likely to play the "*Adjudicator*" role in the "*Convict*" event. As a result, all the three models can predict the role type correctly. Therefore, the first two instances demonstrate that entity types can provide useful cues in predicting frequent participating role types.

For Instance (3), the entity type and role type of the event argument candidate "*former WorldCom CEO Bernard Ebbers*" are "*Individual*" and "*Recipient*" respectively in the "*Transfer-Money*" event triggered by "*loan*". As the role types "*Giver*" and "*Recipient*" are often associated with the same entity type "*Individual*" in the "*Transfer-Money*" event, DMBERT fails to distinguish "*Giver*" and "*Recipient*". However, DMBERT (CL) and METOR can recognize the role type correctly. Both models benefit from contrastive learning. It helps to avoid the confusion between role types with the same entity type in the same event. Instance (4) shows another similar example.

In Instance (5), the entity type and role type of the event argument candidate "*how many people*" are "*Indeterminate*" and "*Target*" respectively. Semantically, if an event argument candidate represents an "*Indeterminate*" object, it is highly likely that the object does not play any role in the events. This strong semantic coupling causes DMBERT to predict wrongly on the role type. DMBERT (CL) uses the vanilla contrastive learning to avoid entity type overdependency of "*Indeterminate*" but fails when most event argument candidates with entity type "*Indeterminate*" do not play any role in the training set. However, METOR avoids such errors as it has reduced the entity type overdependency problem and learned more semantic information besides entity types.

For Instance (6), the entity type and role type of the event argument candidate "*the space shuttle Columbia*" are "*Air*" and "*None*" respectively in the "*Die*" event triggered by "*killing*". Similarly, if an event argument candidate represents an "*Air*" object, it is highly likely that the object plays the "*Instrument*" role. This hinders DMBERT and DMBERT (CL) from understanding the semantic information from the text and causes them to make the wrong prediction. Our proposed METOR model can predict the role type correctly. Therefore, METOR can reduce false negative prediction (as shown in Instance (5)) as well as false positive prediction (as shown in Instance (6)), thus achieving promising EAE performance.

6. Conclusion

In this paper, we study entity type dependency for EAE and conduct experiments to evaluate its effects on the overall performance based on different EAE encoders. Experimental analysis shows that different EAE encoders have suffered from varying degrees of entity type overdependency, which degrades the overall performance. To tackle entity type overdependency, we propose a novel Multi-view Entity Type Overdependency Reduction (METOR) model. In the proposed model, two novel contrastive learning methods are proposed to reduce entity type overdependency from both positive samples and negative samples, and a cyclic training strategy is designed to enable the two contrastive learning methods to collaborate with each other efficiently. Experimental results based on the ACE 2005 English dataset have shown that our proposed METOR model has outperformed the baseline models and achieved the state-of-the-art performance for EAE.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work was supported by National Key Research and Development Program of China [Grant No. 2020YFC0833402] and National Natural Science Foundation of China [Grant Nos. 61976021]. This work was also supported by the financial support (No. 202206030113) by China Scholarship Council during a visit to Nanyang Technological University.

References

[1] Y. Chen, L. Xu, K. Liu, D. Zeng, J. Zhao, Event extraction via dynamic multi-pooling convolutional neural networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015, pp. 167–176.

[2] X. Wang, Z. Wang, X. Han, Z. Liu, J. Li, P. Li, M. Sun, J. Zhou, X. Ren, HMEAE: Hierarchical modular event argument extraction, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 5777–5783.

[3] X. Xiangyu, W. Ye, S. Zhang, Q. Wang, H. Jiang, W. Wu, Capturing event argument interaction via a bi-directional entity-level recurrent decoder, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 210–219.

[4] H. Peng, T. Gao, X. Han, Y. Lin, P. Li, Z. Liu, M. Sun, J. Zhou, Learning from context or names? an empirical study on neural relation extraction, 2020, arXiv preprint arXiv:2010.01923.

[5] X. Wang, X. Han, Z. Liu, M. Sun, P. Li, Adversarial training for weakly supervised event detection, 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), 2019, pp. 998–1008.

[6] Y. Hong, J. Zhang, B. Ma, J. Yao, G. Zhou, Q. Zhu, Using cross-entity inference to improve event extraction, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 1127–1136.

[7] X. Liu, Z. Luo, H.-Y. Huang, Jointly multiple events extraction via attention-based graph information aggregation, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1247–1256.

[8] T.H. Nguyen, K. Cho, R. Grishman, Joint event extraction via recurrent neural networks, in: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 300–309.

[9] T. Chen, S. Kornblith, M. Norouzi, G. Hinton, A simple framework for contrastive learning of visual representations, in: International Conference on Machine Learning, PMLR, 2020, pp. 1597–1607.

[10] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick, Momentum contrast for unsupervised visual representation learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE Computer Society, 2020, pp. 9726–9735.

[11] H. Fang, S. Wang, M. Zhou, J. Ding, P. Xie, Cert: Contrastive self-supervised learning for language understanding, 2020, arXiv preprint arXiv:2005.12766.

[12] D. Wang, N. Ding, P. Li, H. Zheng, CLINE: Contrastive learning with semantic negative examples for natural language understanding, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 2332–2342.
[13] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, Supervised contrastive learning, Adv. Neural Inf. Process. Syst. 33 (2020).
[14] H. Liu, F. Zhang, X. Zhang, S. Zhao, X. Zhang, An explicit-joint and supervised-contrastive learning framework for few-shot intent classification and slot filling, in: Findings of the Association for Computational Linguistics: EMNLP 2021, 2021, pp. 1945–1955.
[15] C. Yang, J. Zou, J. Wu, H. Xu, S. Fan, Supervised contrastive learning for recommendation, Knowl.-Based Syst. 258 (2022) 109973.
[16] S. Liao, R. Grishman, Using document level cross-event inference to improve event extraction, in: Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, 2010, pp. 789–797.
[17] S. Riedel, A. McCallum, Fast and robust joint models for biomedical event extraction, in: Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing, 2011, pp. 1–12.
[18] Q. Li, H. Ji, L. Huang, Joint event extraction via structured prediction with global features, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 73–82.
[19] R. Huang, E. Riloff, Modeling textual cohesion for event extraction, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 26, (1) 2012.
[20] L. Sha, J. Liu, C.-Y. Lin, S. Li, B. Chang, Z. Sui, Rbpb: Regularization-based pattern balancing method for event extraction, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 1224–1234.
[21] L. Sha, F. Qian, B. Chang, Z. Sui, Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.
[22] S. Yang, D. Feng, L. Qiao, Z. Kan, D. Li, Exploring pre-trained language models for event extraction and generation, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 5284–5294.
[23] Y. Lu, H. Lin, J. Xu, X. Han, J. Tang, A. Li, L. Sun, M. Liao, S. Chen, Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 2795–2806.
[24] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (2020) 1–67.
[25] Y. Ma, Z. Wang, Y. Cao, M. Li, M. Chen, K. Wang, J. Shao, Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction, 2022, arXiv preprint arXiv:2202.12109.
[26] X. Du, C. Cardie, Event extraction by answering (almost) natural questions, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 671–683.
[27] J. Liu, Y. Chen, K. Liu, W. Bi, X. Liu, Event extraction as machine reading comprehension, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 1641–1651.
[28] B. Gunel, J. Du, A. Conneau, V. Stoyanov, Supervised contrastive learning for pre-trained language model fine-tuning, in: International Conference on Learning Representations, 2020.
[29] V. Suresh, D. Ong, Not all negatives are equal: Label-aware contrastive loss for fine-grained text classification, in: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, 2021, pp. 4381–4394.
[30] Z. Wang, X. Wang, X. Han, Y. Lin, L. Hou, Z. Liu, P. Li, J. Li, J. Zhou, CLEVE: Contrastive pre-training for event extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6283–6297.
[31] S. Ohashi, J. Takayama, T. Kajiwara, C. Chu, Y. Arase, Text classification with negative supervision, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 351–357.
[32] Y. Lu, H. Lin, X. Han, L. Sun, Distilling discrimination and generalization knowledge for event detection via delta-representation learning, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 4366–4376.
[33] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, 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), 2019, pp. 4171–4186.
[34] G. Salton, M.J. McGill, Introduction to Modern Information Retrieval, 1986.
[35] S. Yao, K. Shuang, R. Li, S. Su, FGCAN: Filter-based gated contextual attention network for event detection, Knowl.-Based Syst. 228 (2021) 107294.
[36] J. Lv, Z. Zhang, L. Jin, S. Li, X. Li, G. Xu, X. Sun, Trigger is non-central: Jointly event extraction via label-aware representations with multi-task learning, Knowl.-Based Syst. 252 (2022) 109480.
[37] T. Dozat, C.D. Manning, Deep biaffine attention for neural dependency parsing, 2016, arXiv preprint arXiv:1611.01734.
[38] D. Wadden, U. Wennberg, Y. Luan, H. Hajishirzi, 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), 2019, pp. 5784–5789.
[39] S. Li, H. Ji, J. Han, Document-level event argument extraction by conditional generation, in: Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 894–908.
[40] I.-H. Hsu, K.-H. Huang, E. Boschee, S. Miller, P. Natarajan, K.-W. Chang, N. Peng, DEGREE: A data-efficient generation-based event extraction model, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 1890–1908.
[41] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, L. Zettlemoyer, BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7871–7880.
[42] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv. Neural Inf. Process. Syst. 30 (2017).