applied sciences

MDPI

Article

Short Text Event Coreference Resolution Based on Context Prediction

Xinyou Yong [1,2,†], Chongqing Zeng [1,2,†], Lican Dai [1], Wanli Liu [1] and Shimin Cai [2,*]

[1] Second Laboratory, Southwest Institute of Electronic Technology, Chengdu 610036, China; yongxinyou960218@163.com (X.Y.); zengchongq@126.com (C.Z.); licand@mail.ustc.edu.cn (L.D.); liu.wli@163.com (W.L.)

[2] School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

* Correspondence: shimincai@uestc.edu.cn

† These authors contributed equally to this work.

Abstract: Event coreference resolution is the task of clustering event mentions that refer to the same entity or situation in text and performing operations like linking, information completion, and validation. Existing methods model this task as a text similarity problem, focusing solely on semantic information, neglecting key features like event trigger words and subject. In this paper, we introduce the event coreference resolution based on context prediction (ECR-CP) as an alternative to traditional methods. ECR-CP treats the task as sentence-level relationship prediction, examining if two event descriptions can create a continuous sentence-level connection to identify coreference. We enhance ECR-CP with a fusion coding model (ECR-CP$^+$) to incorporate event-specific structure and semantics. The model identifies key text information such as trigger words, argument roles, event types, and tenses via an event extraction module, integrating them into the encoding process as auxiliary features. Extensive experiments on the benchmark CCKS 2021 dataset demonstrate that ECR-CP and ECR-CP$^+$ outperform existing methods in terms of precision, recall, and F1 Score, indicating their superior performance.

Keywords: coreference resolution; relationship prediction; universal information extraction; deep learning

check for updates

Citation: Yong, X.; Zeng, C.; Dai, L.; Liu, W.; Cai, S. Short Text Event Coreference Resolution Based on Context Prediction. *Appl. Sci.* **2024**, *14*, 527. https://doi.org/10.3390/app14020527

Academic Editors: Juan Pavón and Yoshiyasu Takefuji

Received: 26 September 2023
Revised: 3 December 2023
Accepted: 5 January 2024
Published: 7 January 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Event coreference resolution (ECR) is a crucial natural language processing task that involves identifying and clustering together different textual mentions of the same event. The task holds significant importance in enabling a variety of downstream applications, including information extraction, question answering, text summarization, etc. [1,2]. For example, in the application of information extraction, ECR can help to build more coherent and complete knowledge graphs or databases by linking different mentions of the same event and correctly identifying and linking relevant information to answer questions accurately [3]. Figure 1 shows an example of the ECR task, where the input consists of two different segments of text and the output is the binary confidence of event coreference.

ECR is also an important part of constructing event graphs [4]. The entire process is similar to entity linking in knowledge graphs, where identical event nodes in the real world are clustered together, further improving and supplementing the various components of the event and saving them as a new node in the graph structure. Some researchers have modeled the ECR task as a text similarity calculation problem, using neural networks such as CNN (Convolutional Neural Network) [5] and RNN (Recurrent Neural Network) [6] to represent two pieces of text as vectors and determining whether they are coreferential by calculating the similarity between the vectors. With the emergence of pre-training models such as BERT [7], some researchers have used the output of pre-training models

as the vector representation of text and calculated similarity. Others have used Siamese networks [8] to classify event pairs and determine if they are coreferential.

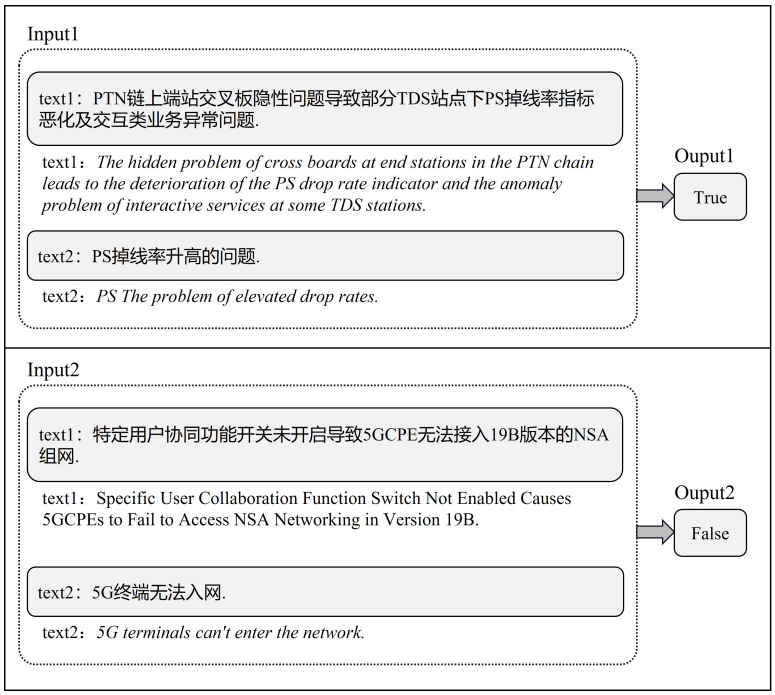


Figure 1. Example illustration of ECR task.

However, training neural networks such as CNN and RNN to represent text as vectors requires training parameters from scratch and is not suitable for small datasets [9]. In contrast, pre-training models only need to be fine-tuned after being pre-trained on a large corpus, which can achieve faster convergence speed when combined with downstream tasks. Additionally, determining event coreference based on whether the similarity reaches a threshold is biased, as there may not be a clear classification boundary for this task, making it difficult to determine the threshold, or there may not be a clear boundary at all. Using BERT for text representation or Siamese networks for text classification also fails to fully utilize the inherent next sentence prediction (NSP). Furthermore, the above techniques only focus on the semantic information of the event description text, ignoring a variety of key features such as event trigger words and event subjects, resulting in missing features.

To address the limitations of existing techniques, we propose a short text event coreference resolution method based on context prediction (referred to ECR-CP). The novelty of ECR-CP is illustrated from three perspectives. In more detail, from a problem-modeling perspective, we model the ECR task as a sentence-level relationship prediction issue by utilizing the NSP inherent in BERT. We consider pairs of events that can form a continuous sentence-level relationship to have coreferential relationships. This is consistent with human language habits, where coherent sentences in everyday conversations often describe the same fact. From a feature extraction perspective, we extract key information such as trigger words, argument roles, event types, and tense from event extraction and incorporate them as auxiliary features to improve the accuracy. From the algorithm performance perspective, BERT-based ECR-CP has a smaller training cost and achieves better performance in comparison to the other benchmark methods based on neural networks.

In summary, the main contributions of our work are as follows:

- We propose a novel short text event coreference resolution method based on context prediction (referred to ECR-CP). ECR-CP takes the ECR task as a discourse relation prediction issue and determines whether two event descriptions refer to the same event based on whether they construct a coherent discourse relation.

- We further improve ECR-CP with a fusion-coding model (referred to ECR-CP$^+$) by integrating diverse features of the special structure of events, as well as semantic information, and unifying the encoding of these features and textual content to further enhance the accuracy of ECR.
- According to the experiments conducted on the CCKS 2021 dataset, ECR-CP and ECR-CP$^+$ achieves the best performance in terms of precision, recall, and F1 score in comparison with these benchmark and state-of-the-art methods.

2. Related Works

ECR is one of the key subtasks in event extraction and fusion. However, early coreference resolution tasks mainly focused on the entity level. In the domain of entity coreference resolution, Kejriwal et al. [10] proposed an unsupervised algorithm pipeline for learning Disjunctive Normal Form (DNF) blocking schemes on Knowledge Graphs (KGs), as well as structurally heterogeneous tables that may not share a common schema. This approach aims to address entity resolution problems by mapping entities to blocks. Additionally, Šteflovič et al. [11] aim to enhance classifier performance metrics by incorporating the results of entity coreference analysis into the data preparation process for classification tasks.

Due to the complexity of events themselves, as well as a lack of relevant language resources, research on ECR both domestically and abroad started relatively late and has developed more slowly than event extraction techniques. The ACE2005 corpus [12] was the first to add coreference attributes to event information, and in 2015, the Knowledge Base Population (KBP) [13] began related evaluation tasks, laying the foundation for subsequent research.

Early research on ECR was mainly based on rule-based methods [14–18], but with the widespread application of machine learning methods. Recent research has mainly used traditional machine learning and neural network methods to complete ECR tasks [1]. A binary classifier is trained to determine whether two pieces of text refer to the same real-world event, and then coreferential events are optimized by clustering [19]. The differences in various methods mainly lie in the problem modeling, text feature extraction methods, and the internal details of the network model. Depending on whether manual annotation of language resources is required, ECR methods can be divided into three categories: supervised, semi-supervised, and unsupervised learning.

Among them, supervised learning is the earliest and most widely used research approach. For example, Fang et al. [20] designed a multi-layer CNN to extract event features, obtained deep semantic information, and further improved the performance of the coreference resolution algorithm by using multiple attention mechanisms. Dai et al. [21] enhanced the representation of event text features using a Siamese network framework and used the Circle Loss loss function to maximize intra-class event similarity and minimize inter-class event similarity. Liu et al. [22] trained a support vector machine (SVM) classifier based on more than 100 event features to determine whether the events refer to the same entity. Another attempt is made to utilize large amounts of out-of-domain text data.

Due to the high cost of manual annotation, many scholars have attempted to use semi-supervised methods to study ECR tasks. The main idea is to use a small amount of labeled data to construct a learning algorithm to learn data distribution and features, thus completing the labeling of unlabeled samples. For example, Sachan et al. [23] achieved good results on a small-scale training dataset by using active learning to select information-rich instances. Similarly, Chen et al. [24] employed active learning to select informative instances, indicating that only a small number of training sentences need to be annotated to achieve state-of-the-art performance in event coreference. Another attempt is made to utilize large amounts of out-of-domain text data [25].

Unsupervised learning methods completely eliminate the dependence on labeled data and are often probabilistic generative models. For example, Bejan et al. [26] constructed a generative, parameter-free Bayesian model based on hierarchical Dirichlet processes and infinite factorial hidden Markov models to achieve unsupervised ECR task learning.

Chen et al. [27] addressed the relatively scarce research on an unsupervised Chinese event coreference resolution task by proposing a generative model. When evaluated on the ACE 2005 corpus, the performance of this model was comparable to that of supervised tasks.

In addition, the differences between various methods also lie in the scoring processing steps of coreference relationships, which can be divided into two models: event-pair models and event-ranking models [28,29]. The event-pair model [30–32] is a binary classification model that independently determines whether each event pair refers to the same entity and then aggregates events that refer to each other to form coreferential event clusters. The event-ranking model judges the current event's coreference with all other candidate events and obtains a ranking result based on the degree of coreference, then divides the events into clusters according to a set threshold λ.

3. Model

From a problem-modeling perspective, ECR-CP models the ECR task as the issue of predicting the relationship between the upper and lower sentences. And, it is further enhanced to improve the model accuracy with a fusion-coding model that incorporates event features into the encoding process. In predicting the coherence of two event descriptions in forming a continuous sentence, we use the BERT-WWM-Chinese model (referred to BERT), which uses other pre-training models based on the transformer–encoder structure, to incorporate the NSP. However, the NSP does not actually predict the next sentence, but rather determines whether the sentence pairs are adjacent. This aligns with the original intention of ECR-CP, which is to judge coreference by predicting whether two event descriptions can form a coherent sentence.

Regarding the construction of event features based on the fusion coding model, we selected three parts as features: event type, trigger words and nearby context, and event arguments. The reason is that if two event description texts are coreferential, the event types they contain should be consistent, making the event category an important feature for judging coreference. Secondly, trigger words are important indicators of event occurrence, and considering the length of the text, the trigger words and the five characters before and after them were further selected as another important feature. Finally, information about event arguments, such as participants, time, and location, can also reflect to some extent whether events are coreferential, as well as it greatly simplifies the expression of complex events. Therefore, event arguments are also input as auxiliary features for classification into the model.

This section first introduces the overall model architecture of ECR-CP and ECR-CP$^+$, then describes its components, including the input layer, BERT-based encoding layer, and output layer.

3.1. Model Architecture

The overall model architecture of ECR-CP and ECR-CP$^+$ are shown in Figure 2. The flowcharts of ECR-CP and ECR-CP$^+$ are similar, and its significant difference is whether the input layer includes the fusion-coding model. ECR-CP directly uses two event description texts to construct the BERT input vector, while ECR-CP$^+$ use the fusion-coding model to unify the encoding of event features and text content as the BERT input vector. Regarding the fusion process of text semantics and event features in the fusion coding model, two event description texts are firstly input into the event extraction module based the universal information extraction model (UIE) [33] to obtain the event type, trigger word and context, and event arguments for each of the two events, and then these event features with the original text are concatenated based on the feature fusion module, which are separated by the "[unused]" special classification token. Note that UIE is based on the pre-training model ERNIE, which allows for the extraction of event features by inputting schema and text. The BERT input vector in the input layer is taken as the input of the BERT-based encoder layer for the upper and lower sentences in BERT. In the BERT-based encoder layer, the internal multiple layers of the Transformer encode the input to obtain a vector representation for

each character that fuses the contextual semantics, and the vector representation of the entire text (i.e., BERT output vector) is obtained using an average pooling strategy. Finally, the BERT output vector is input into the fully connected layer classifier (i.e., DenseNet) to obtain the confidence of event coreference. Herein, we use BERT that has been pre-trained on a larger dataset and fine-tune it on the CCKS 2021 dataset. Thus, the ECR-CP can effectively handle the challenge of small datasets in the ECR task

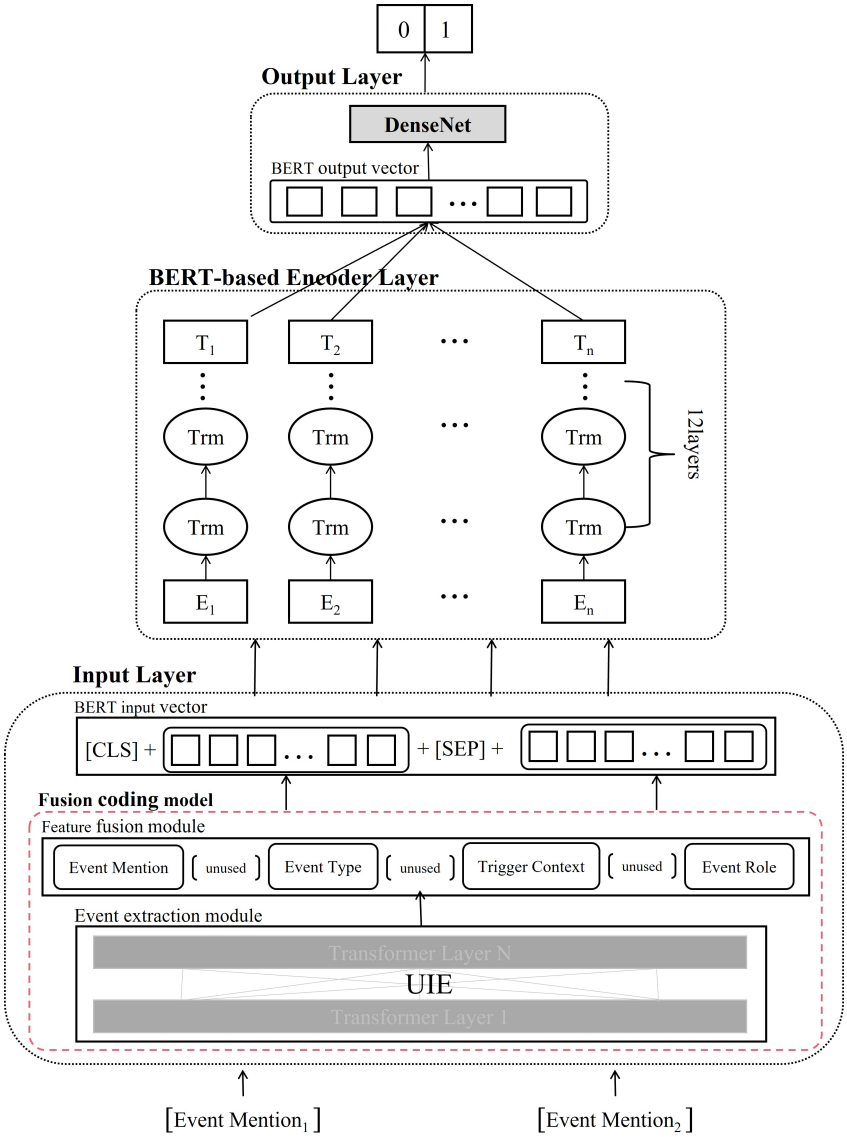


Figure 2. Model architecture of ECR-CP with the fusion coding model (i.e., ECR-CP$^+$).

3.2. Input Layer

The construction of the BERT input vector in ECR-CP is simply concatenated with two event description texts. In this subsection, we concretely illustrate the fusion coding model in ECR-CP$^+$. The fusion coding model is similar in input form to the Prompt paradigm [34], which has been hotly researched in recent years, and originally refers to the design of input forms or templates for downstream tasks so that the language model can better recall what it learned during pre-training. The core idea of the Prompt paradigm is to give the model some hints so that it can better perform the downstream task. The fusion-coding model adds the key information of the event directly to the input layer and uses additional text in the input segment to prompt the model about whether the event is coreferential or not.

Specifically, as shown in Figure 2, two event description texts are input to the fusion-coding model, and the event category, trigger word and context, and argument are obtained by the event extraction module-based UIE, then directly spliced with the source text based on the feature fusion module, separated by the [unused] identifier, to obtain two new text inputs, i.e., the new text retains the original event description and adds the event features. Then, the two new texts are used as the input of the BERT-based encoder layer, and the input form is consistent with the event feature's independent encoding process.

3.3. BERT-Based Encoder Layer

The BERT-based encoder layer is generally in a unified manner. The BERT input vector is used as the input sequence

$$X = \{x_1, x_2, \ldots, x_n\} \tag{1}$$

the embedding representation of each word incorporating semantic information, location information, and sentence information is obtained by Formula (1),

$$E = \{e_1, e_2, \ldots, e_n\}. \tag{2}$$

The encoding process goes through two main parts of BERT: the Transformer encoder module and the pooling module (total 12 layers). Specifically, each encoder layer in the Transformer encoder module contains a self-attention mechanism and a feedforward neural network. Among them, the self-attention mechanism calculates an attention weight matrix A_i. Based on the embedding representation e_i of each word in the input sequence, we interact the embedding representation of each word with the embedding representations of other words. Then, the feedforward neural network will perform a non-linear transformation on the interacted embedding representation to obtain the encoded embedding representation H_i,

$$H = TransformerEncoder(E). \tag{3}$$

Within the final layer of BERT, the pooling module is applied to transform the encoded embedded representation into a fixed-dimension vector representation T_i. Two specific pooling techniques, namely average pooling and max pooling, can be employed. Subsequently, the results obtained from these pooling operations are concatenated to yield the ultimate vector representation C (i.e., BERT output vector),

$$C = Concatenate([AvgPooling(H), MaxPooling(H)]) \tag{4}$$

In the given equations, the term TransformerEncoder represents the Transformer encoder, while AvgPooling and MaxPooling correspond to average pooling and max pooling operations, respectively. Concatenate denotes the vector concatenation operation. The ultimate textual vector representation C obtained from this process can be effectively employed in a range of downstream natural language processing tasks.

3.4. Output Layer

In the output layer, we assume the BERT output vector as $C \in \mathbb{R}^{d_1}$ for ECR-CP (or ECR-CP$^+$) is directly input into DenseNet. DenseNet takes the input $h = C \in \mathbb{R}^{d_1}$ and feeds it into the dense blocks with the ReLU activation function [35]. The dense blocks output the feature vectors V_h, which are finally concatenated using the pooling layer. We use the classification layer with the Sigmoid function to calculate the co-reference confidence score p. The detailed process is simply outlined in Equation (5).

$$\begin{aligned} V_h &= Relu(W_h \cdot h + b_h) \\ p &= Sigmoid(W_{\text{out}} \cdot V_h) + b_{\text{out}} \end{aligned} \tag{5}$$

In the aforementioned equation, W_h, b_h, W_{out} and b_{out} are parameters in DenseNet.

4. Experimental Setup

4.1. Datasets

In our experiments, we used the Chinese ECR dataset released during the 2021 National Conference on Knowledge Graph and Semantic Computing (CCKS 2021), which is specifically designed for the communication domain. In the CCKS 2021 dataset, there are various process-related knowledge in the communication domain, such as hardware installation, parameter configuration, integrated debugging, fault hading, etc. The tasks of event extraction and event coreferential resolution are important means to achieve the sorting of fault context, troubleshooting steps, and recovery steps. The corpus in the CCKS 2021 dataset consists of publicly accessible fault handling cases from Huawei, incorporating a multitude of complex domain-specific terms, ambiguous events, shared elements, and diverse representations of event elements.

There are eight event types including "Index Fault", "Soft Hardware fault", "Collect Data", "Check", "Setting Fault", "External Fault", "Set Machine", and "Operate Machine". The event-specific trigger terms and event elements are diverse, such as "Configuration", "Mismatch", and "Inconsistent" in the "Set Machine" event. Meanwhile, the event descriptions are presented in the form of text pairs, accompanied by labels denoting "True" or "False" to indicate whether they are coreferent. Thus, the CCKS 2021 dataset can concurrently evaluate the tasks of event extraction and event coreferential resolution, and we can use the results of event extraction to supplement the auxiliary features in the task of event coreferential resolution. The CCKS 2021 dataset is open access and contains a total of 15,000 sample data which were randomly divided into training, validation, and test sets with a ratio of 6:2:2.

4.2. Evaluation Metrics

The accuracy of ECR models can be evaluated from two perspectives. One is to use precision (P), recall (R), and F1 Score in a simple binary classification model. The other is to evaluate the algorithm performance based on the standard coreference chain or event set. As we view event pair coreference resolution as a binary classification task, we use precision, recall, and F1 score to evaluate ECR models. We first compute the confusion matrix of classification results. It includes four values according to the predicted class and actual class; the true positive rate (TP), which describes the consistency of positive examples; the true negative rate (TN), which describes the consistency of negative examples; the false positive rate (FP), which describes that the positive examples are incorrectly labeled as the negative examples; and the false negative rate (FN), which describes that the negative examples are incorrectly labeled as the positive examples. Then, to keep our description as self-contained as possible, we present these indexes briefly.

Precision measures the prediction performance, which is the percentage of the positive examples among the predicted results. Its specific formula is as follows,

$$Precision = \frac{TP}{TP + FP}. \tag{6}$$

Recall measures the prediction performance, which is the percentage of the positive examples correctly predicted among all predicated positive examples. Its specific formula is as follows,

$$Recall = \frac{TP}{TP + FN}. \tag{7}$$

Note that the indexes of recall and sensitivity are equal in the formula.

F_1 score is generally defined in mathematics to reconcile the results of precision and recall. Its specific formula is as follows,

$$F_1 - Score = 2 \cdot \frac{precision \cdot recall}{precision + recall}. \tag{8}$$

4.3. Implementation Details

We used the BERT-WWM-Chinese model to predict whether two event descriptions can construct a coherent discourse relation. In terms of experimental parameters, we set the maximum input text length to 256 by calculating the average text length of the training samples. We used a batch size of 32 and adopted the average pooling strategy for pooling. We used 12 layers of the Transformer and employed 12 multi-head attention mechanisms, with GeLU as the activation function for the hidden layers. To prevent overfitting, we set the dropout probability of the hidden layer to 0.1. The learning rate was set to 1×10^{-5}, and the number of training epochs was set to five to reducing the training time. Note that we empirically obtain the convergence of the loss rate before five training epochs. After each third of the training epochs, we conducted validation and saved the model parameters with the highest F1 score on the validation set for the final test.

5. Experimental Results and Discussion

Table 1 shows the performance of ECR-CP and ECR-CP$^+$ in the ECR task. We can see that the model performance demonstrates that modeling ECR as a discourse relation prediction task is feasible. By comparing the results of ECR-CP and ECR-CP$^+$, it can be observed that incorporating event features leads to improvements in precision and recall of the model, with recall showing a particularly significant improvement. ECR-CP$^+$ showed an improvement of 2.2% in precision, 7.9% in recall, and a 5.2% increase in F1 Score after incorporating the event features.

Table 1. The performance of ECR-CP and ECR-CP$^+$ on the test set.

Methods	P (%)	R (%)	F1-Score (%)
ECR-CP	96.3	89.0	92.5
ECR-CP$^+$	98.5 (2.2)	96.9 (7.9)	97.7 (+5.2)

This can be attributed to the fact that the determination of event co-reference heavily relies on the consistency of various entities involved in the events. Excessive occurrence of irrelevant content in the event description text may hinder the model's ability to capture crucial information. Consequently, without the explicit event feature "Prompt", the model tends to classify certain co-reference examples as non-coreferential. However, the introduction of event features significantly enhances the model's capability to identify positive instances.

Furthermore, we conduct comparative experiments between several existing benchmark and state-of-the-art event-pair-based ECR methods and our models on the CCKS2021 dataset. The methods used in the comparative experiment are as follows:

CNNs-SVM is a straightforward ECR method based on semantic similarity. Firstly, it applies convolution operations to the text matrix using multiple convolutional kernels of different sizes. Then, the outputs of each kernel are pooled and the concatenated output from different kernels forms the vector representation of the text. The textual vectors are taken as the input features of the support vector machine (SVM). Based on the training set, we can train the best SVM-based binary classification for the ECR task and determine the textual vectors of events. The classification threshold σ of the co-reference between event pairs is determined by computing the cosine similarity between their respective textual vectors. If the cosine similarity of event pairs in the testing set is greater than σ, the event pairs are considered coreferent.

Sentence-BERT is a network model proposed by Reimers et al. [36] for addressing semantic matching tasks. It is a widely adopted approach for computing semantic similarity. Sentence-BERT utilizes a Sentence-BERT model, similar to Siamese networks, to obtain vector representations u and v for two text segments. These vectors are then concatenated as $[u, v, |u - v|]$, and passed through fully connected layers. The SoftMax classifier is employed for text semantic matching, specifically determining if the events are coreferent. More details of Sentence-BERT can be learned according to [36].

CorefNet is an ECR model based on a convolutional neural network with a multi-attention mechanism, proposed by Fang et al. [20]. It leverages multiple layers of convolutional operations to extract text features, utilizes dot-product attention to discover shared features in two events, and incorporates a self-attention mechanism to capture important features within sentences. Additionally, it integrates semantic role features to improve the model's performance. The more details of CorefNet can be learned according to [20].

EPASE is an ECR model proposed by Zeng et al. [31], which utilizes Event-specific Paraphrases and Argument-aware Semantic Embeddings. The paraphrases' embedding module aims to identify different expressions denoting the same event across different sentences. Meanwhile, the argument-aware semantic embedding module incorporates event arguments, such as subject and object, to differentiate between distinct event descriptions. More details of EPASE can be learned according to [31].

Lemma Heuristics (LH), proposed by Ahmed et al. [37], is an ECR method that decomposes the ECR problem into two parts: (a) an effective heuristic approach for filtering a large number of non-coreferent mention pairs, and (b) a training method for balancing sets of coreferent and non-coreferent mention pairs. This significantly reduces computational requirements, enabling the algorithm to learn coreference beyond surface matching and effectively improve the model training results. The more details of LH can be learned according to [37].

Table 2 presents the experimental results of various methods on the CCKS 2021 dataset for ECR. It is evident that ECR-CP$^+$, which utilizes event feature fusion encoding, achieves optimal performance in terms of precision, recall, and F1 Score, achieving scores of 98.5%, 96.9%, and 97.7% respectively. Compared to the best-performance of existing methods in three evaluation metrics, ECR-CP$^+$ shows improvements of 1.0%, 0.4%, and 2.9% in these respective metrics.

Table 2. The experimental results of various methods for ECR based on the CCKS 2021 dataset. The best results are indicated in bold and the second ones are highlighted with underlines.

Methods	P (%)	R (%)	F1-Score (%)
CNNs-SVM	72.3	71.7	72.0
Sentence-BERT	97.1	91.2	94.1
CorefNet	92.5	89.1	90.8
EPASE	<u>97.5</u>	92.3	<u>94.8</u>
LH	90.5	<u>96.5</u>	93.4
ECR-CP (Ours)	96.3	89.0	92.5
ECR-CP$^+$ (Ours)	**98.5**	**96.9**	**97.7**

However, both the CNNs-SVM model, which employs traditional CNN for text feature extraction, and CorefNet demonstrate subpar performance. This could be attributed to the difficulty in establishing distinct classification boundaries among text vectors, resulting in the suboptimal effectiveness of using machine learning to determine similarity thresholds. Although CorefNet benefits from multiple attention mechanisms and incorporates semantic role features, its performance falls short of other BERT-based models due to limitations in traditional neural network feature extraction capabilities. Furthermore, Sentence-BERT, EPASE, and LH exhibit better performance than ECR-CP, but fall short compared to ECR-CP$^{[+]}$ that incorporates event features. This suggests that modeling the ECR task as a sentence-pair prediction is effective, and the inclusion of event features significantly enhances the model's recall for coreferent events.

Based on a comprehensive analysis, the enhancement in the performance of ECR-CP$^+$ can be primarily attributed to three factors:

(1) By modeling ECR as a sentence-pair prediction task, a single model is capable of extracting features from both event texts, which facilitates information interaction and enables the language model to more effectively identify the shared characteristics and distinctions among event pairs.

(2) Utilizing pre-trained models for feature extraction offers enhanced semantic capturing capabilities in contrast to traditional CNNs or custom-designed convolutional neural networks. Moreover, with the inclusion of the multi-head attention mechanism, it assigns greater weights to words that exert substantial influence on the automatic coreference outcomes.

(3) The inclusion of event features, encompassing event type, trigger words, and arguments, to a certain extent, reflects the coreferential nature of events, particularly when dealing with semantically intricate event descriptions. Introducing event features notably improves the model's discernment of positive instances.

6. Conclusions

We propose ECR-CP to address the problem of the event pair coreference resolution. Compared with traditional probability-based and graph-based models, ECR-CP models ECR as a discourse relation prediction task which predicts whether two event mentions can construct a coherent discourse relation and thereby determine whether the event mentions are coreferent. Furthermore, incorporating event features further improves the model performance, so we proposed ECR-CP$^+$ with event feature fusion encoding. The experimental results show evidence that the event features are significant for improving the model accuracy. Finally, we demonstrated the superiority of our models via extensive experiments in comparison with these existing benchmark and state-of-the-art methods.

In the current work, we used the large pre-trained models based on deep Transformer structures. The large number of parameters and layers can result in slow computation speeds. Therefore, we should consider reducing the number of layers or incorporating knowledge distillation to reduce the number of parameters in our future work.

Author Contributions: Conceptualization, X.Y., C.Z., and S.C.; methodology, X.Y. and C.Z.; software, C.Z. and L.D.; validation, X.Y., W.L., and S.C.; investigation, X.Y., C.Z., and S.C.; data curation, X.Y. and C.Z.; writing—original draft preparation, X.Y., C.Z., and S.C.; writing—review and editing, all authors; supervision, S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Ministry of Education of Humanities and Social Science Project under Grant No. 21JZD055.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CCKS 2021 dataset is open source for technique evaluation tasks in the 2021 China Conference on Knowledge Graph and Semantic Computing. The access of this dataset requires contact with the correspondence author. The data are not publicly available due to the close of open access.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Liu, R.; Mao, R.; Luu, A.T.; Cambria, E. A brief survey on recent advances in coreference resolution. *Artif. Intell. Rev.* **2023**, *56*, 14439–14481. [CrossRef]
2. Chen, L.C.; Chang, K.H. An Extended AHP-Based Corpus Assessment Approach for Handling Keyword Ranking of NLP: An Example of COVID-19 Corpus Data. *Axioms* **2023**, *12*, 740. [CrossRef]
3. Yang, Y.; Wu, Z.; Yang, Y.; Lian, S.; Guo, F.; Wang, Z. A survey of information extraction based on deep learning. *Appl. Sci.* **2022**, *12*, 9691. [CrossRef]
4. Hu, Z.; Jin, X.; Chen, J.; Huang, G. Construction, reasoning and applications of event graphs. *Big Data Res.* **2021**, *7*, 80–96.
5. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
6. Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Phys. D Nonlinear Phenom.* **2020**, *404*, 132306. [CrossRef]
7. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.

8. Chopra, S.; Hadsell, R.; LeCun, Y. Learning a similarity metric discriminatively, with application to face verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–26 June 2005; Volume 1, pp. 539–546.
9. Mohammed, A.; Kora, R. A comprehensive review on ensemble deep learning: Opportunities and challenges. *J. King Saud-Univ.-Comput. Inf. Sci.* **2023**, *35*, 757–774. [CrossRef]
10. Šteflovič, K.; Kapusta, J. Coreference Resolution for Improving Performance Measures of Classification Tasks. *Appl. Sci.* **2023**, *13*, 9272. [CrossRef]
11. Kejriwal, M. Unsupervised DNF Blocking for Efficient Linking of Knowledge Graphs and Tables. *Information* **2021**, *12*, 134. [CrossRef]
12. Walker, C.; Strassel, S.; Medero, J.; Maeda, K. ACE 2005 multilingual training corpus. *Phila. Linguist. Data Consort.* **2006**, *57*, 45.
13. Mitamura, T.; Liu, Z.; Hovy, E.H. Overview of TAC KBP 2015 Event Nugget Track. In Proceedings of the TAC, Gaithersburg, MR, USA, 12–13 November 2015.
14. Hobbs, J.R. Resolving pronoun references. *Lingua* **1978**, *44*, 311–338. [CrossRef]
15. Humphreys, K.; Gaizauskas, R.; Azzam, S. Event coreference for information extraction. In Proceedings of the Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts, Madrid, Spain, 11 July 1997.
16. Glavaš, G.; Šnajder, J. Exploring coreference uncertainty of generically extracted event mentions. In *Proceedings of the International Conference on Intelligent Text Processing and Computational Linguistics*; Springer: Cham, Switzerland, 2013; pp. 408–422.
17. Li, L.; Jin, L.; Jiang, Z.; Zhang, J.; Huang, D. Coreference resolution in biomedical texts. In Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Belfast, UK, 2–5 November 2014; pp. 12–14.
18. Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; Dyer, C. Neural architectures for named entity recognition. *arXiv* **2016**, arXiv:1603.01360.
19. Prasad, K.V.; Vaidya, H.; Rajashekhar, C.; Karekal, K.S.; Sali, R. Automated neural network forecast of PM concentration. *Int. J. Math. Comput. Eng.* **2023**, *1*, 67–78. [CrossRef]
20. Fang, J.; Li, P.; Zhu, Q. Employing Multi-attention Mechanism to Resolve Event Coreference. *Comput. Sci.* **2019**, *46*, 277–281.
21. Dai, B.; Qian, J.; Cheng, S.; Qiao, L.; Li, D. Event Coreference Resolution based on Convolutional Siamese network and Circle Loss. In Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN), Padua, Italy, 18–23 July 2022; pp. 1–7.
22. Liu, Z.; Araki, J.; Hovy, E.H.; Mitamura, T. Supervised Within-Document Event Coreference using Information Propagation. In Proceedings of the 2014 the Language Resources and Evaluation Conference, Reykjavik, Iceland, 26–31 May 2014; pp. 4539–4544.
23. Sachan, M.; Hovy, E.; Xing, E.P. An active learning approach to coreference resolution. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, Buenos Aires, Argentina, 25–31 July 2015.
24. Chen, C.; Ng, V. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30.
25. Peng, H.; Song, Y.; Roth, D. Event detection and co-reference with minimal supervision. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; pp. 392–402.
26. Bejan, C.A.; Harabagiu, S. Unsupervised event coreference resolution. *Comput. Linguist.* **2014**, *40*, 311–347. [CrossRef]
27. Chen, C.; Ng, V. Chinese event coreference resolution: An unsupervised probabilistic model rivaling supervised resolvers. In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, CO, USA, 31 May 2015; pp. 1097–1107.
28. Tran, H.M.; Phung, D.; Nguyen, T.H. Exploiting document structures and cluster consistencies for event coreference resolution. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, Online, 1–6 August 2021; Volume 1.
29. Lu, J.; Ng, V. Constrained multi-task learning for event coreference resolution. 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. 4504–4514.
30. Barhom, S.; Shwartz, V.; Eirew, A.; Bugert, M.; Reimers, N.; Dagan, I. Revisiting joint modeling of cross-document entity and event coreference resolution. *arXiv* **2019**, arXiv:1906.01753.
31. Zeng, Y.; Jin, X.; Guan, S.; Guo, J.; Cheng, X. Event coreference resolution with their paraphrases and argument-aware embeddings. In Proceedings of the 28th International Conference on Computational Linguistics, Online, 8–13 December 2020; pp. 3084–3094.
32. De Langhe, L.; Desot, T.; De Clercq, O.; Hoste, V. A Benchmark for Dutch End-to-End Cross-Document Event Coreference Resolution. *Electronics* **2023**, *12*, 850. [CrossRef]
33. Lu, Y.; Liu, Q.; Dai, D.; Xiao, X.; Lin, H.; Han, X.; Sun, L.; Wu, H. Unified structure generation for universal information extraction. *arXiv* **2022**, arXiv:2203.12277.
34. Liu, P.; Yuan, W.; Fu, J.; Jiang, Z.; Hayashi, H.; Neubig, G. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
35. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.

36. Reimers, N.; Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv* **2019**, arXiv:1908.10084.
37. Ahmed, S.R.; Nath, A.; Martin, J.H.; Krishnaswamy, N. $2 * n$ is better than n^2: Decomposing Event Coreference Resolution into Two Tractable Problems. *arXiv* **2023**, arXiv:2305.05672.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.