Event Coreference Resolution with their Paraphrases and Argument-aware Embeddings

Yutao Zeng, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng
School of Computer Science and Technology, University of Chinese Academy of Sciences;
CAS Key Laboratory of Network Data Science and Technology, Institute of
Computing Technology, Chinese Academy of Sciences
{zengyutao18s,jinxiaolong,guansaiping,guojiafeng,cxq}@ict.ac.cn

Abstract

Event coreference resolution aims to classify all event mentions that refer to the same real-world event into the same group, which is necessary to information aggregation and many downstream applications. To resolve event coreference, existing methods usually calculate the similarities between event mentions and between specific kinds of event arguments. However, they fail to accurately identify paraphrase relations between events and may suffer from error propagation while extracting event components (i.e., event mentions and their arguments). Therefore, we propose a new model based on Event-specific Paraphrases and Argument-aware Semantic Embeddings, thus called EPASE, for event coreference resolution. EPASE recognizes deep paraphrase relations in an event-specific context of sentences and can cover event paraphrases of more situations, bringing about a better generalization. Additionally, the embeddings of argument roles are encoded into event embedding without relying on a fixed number and type of arguments, which results in the better scalability of EPASE. Experiments on both within- and cross-document event coreference demonstrate its consistent and significant superiority compared to existing methods.

1 Introduction

Event coreference resolution clusters event mentions referring to the same real-world event, no matter within a single document (denoted as WD) or across multiple documents (denoted as CD). It is vital for information aggregation and can further benefit many downstream natural language processing applications, including contradiction detection (De Marneffe et al., 2008), text mining (Ferracane et al., 2016) and question answering (Khashabi et al., 2018; Welbl et al., 2018).

Usually, each event consists of an event mention and a few arguments. Take the following two sentences as an example, where event mentions are marked in bold, and the subscripts indicates their IDs:

(1) Perennial party girl Tara Reid **checked**$_{m1}$ herself **into**$_{m_1}$ Promises Treatment Center, her representative **told**$_{m2}$ The Washington Post.

(2) The original trainwreck Tara Reid's publicist **confirmed**$_{m3}$ that the actress Reid was **admitted into**$_{m4}$ Promises Treatment Center in Los Angeles and it was her **decision**$_{m5}$.

these five event mentions in the sentences can be clustered into two sets $\{m1$,$m4$, $m5\}$ and $\{m2$, $m3\}$. Each set constitutes an event coreference chain and the event mentions in it are all coreferential.

An intuitive and effective way to resolve event coreference is to calculate the similarities between event mentions and between their arguments. Many methods (Yang et al., 2015; Choubey and Huang, 2017; Barhom et al., 2019) have adopted these similarities as important features to train classifiers. Essentially, this means is to identify the *paraphrase relation* between events (i.e., one event can be viewed as a paraphrase of another event) by modeling the similarity between event components (i.e., event mentions and their arguments). However, these methods use these event components separately and ignore deep event paraphrase relations within events' context.

In addition, when incorporating event arguments to resolve event coreference, previous methods (Lee et al., 2012; Yang et al., 2015; Choubey and Huang, 2017; Barhom et al., 2019) first extract semantic

Proceedings of the 28th International Conference on Computational Linguistics, pages 3084–3094
Barcelona, Spain (Online), December 8-13, 2020

roles through Semantic Role Labeling (SRL), match them with entity annotations in the corpus to obtain simplified arguments, and finally fill argument slots to get the specified types of arguments. Due to the inconsistency between the output of the SRL system and the entity annotations, they often suffer from error propagation. For instance, in the output of the SRL system, the ARG0 (agent) of **confirmed** ($m3$) is *The original trainwreck Tara Reid's publicist*, but there are two annotated entities *Tara Reid* and *publicist*. It's hard to distinguish which is the right ARG0 using simple matching rules. Besides, a fixed number and type of arguments will also cause poor scalability.

To address these limitations, we propose an Event-specific Paraphrases and Argument-aware Semantic Embeddings enhanced model (**EPASE**) for within- and cross-document event coreference resolution. Here, the event-specific paraphrase, we mean a special kind of paraphrase relation between sentences, only focuses on certain events. For example, when focusing on events **told** (m_2) and **confirmed** (m_3), the above two sentences are paraphrase pairs, but they are not when looking at them from the perspectives of events **checked into** ($m1$) and **confirmed** (m_3). In EPASE, a paraphrase identification module and a special token [COREF] are employed to help the model capture the event-specific paraphrase relations. At the same time, we obtain the argument-aware semantic embeddings of events by incorporating the embeddings of semantic role labels with the token embeddings and aggregating these information through a semantic attention mechanism. Finally, we integrate these two aspects to get the coreference score between events. In general, the contributions of this paper can be summarized as follows:

- We identify event-specific paraphrase relations between sentences by capturing the semantic information of events within the complete sentences, other than incorporating only event components separately. To the best of our knowledge, we are the first to introduce the event-specific paraphrase relations to resolve event coreference.
- We further combine the embeddings of argument roles with the sequence token embeddings to obtain event embeddings within the context. Compared to existing methods, our method requires fewer steps and annotations but achieves better scalability.
- Experiments demonstrate the significant superiority of EPASE, and the event-specific paraphrase features and the argument-aware semantic embeddings are both beneficial to resolving event coreference.

2 Related works

According to the development period and adopted methods, we roughly divide the previous works on event coreference resolution into three categories.

2.1 Feature or template based methods

Early studies (Bejan and Harabagiu, 2010; Lee et al., 2012; Cybulska and Vossen, 2015b) usually adopt lexical features (e.g. string matching features), argument features (e.g. argument alignment features) and basic semantic features (e.g. word embedding similarity and WordNet synonyms) to calculate the similarity between different event mentions. At the same time, some manual rules and event templates are used to check the compatibility between events.

Among them, Cybulska and Vossen (2015b) proposed a model based on event templates. They first set up 5 slots (Action, Time, Location, Human-Participants, Non-Human-Participants), then filled the slots by extracting the five elements of events, and used these slots as the representation of events. The documents were represented as "Bag of Events". Then the documents were clustered, and the compatibility of event slots was checked to determine whether the events in each cluster are coreferential.

2.2 Neural network based methods

Recent studies base on the neural networks to encode event mentions, context, and event arguments to obtain the embeddings of events (Choubey and Huang, 2017; Kenyon-Dean et al., 2018).

For example, Kenyon Dean et al. (2018) proposed an event representation learning model based on neural network. They introduced the *clustering-oriented regularization* term to impel the model to

produce similar embeddings for coreferential events, and dissimilar embeddings otherwise. Their model does not include the preprocessing step of document clustering but encodes the document information as part of event mention embeddings to avoid coreference linking between events in different document clusters.

2.3 Information enhanced methods

More recent studies try to introduce more information through joint modeling or data argumentation methods. The state-of-the-art models (Barhom et al., 2019; Meged et al., 2020) in the past two years have adopted such a strategy to improve the performance of event coreference resolution.

Barhom et al. (2019) presented a method that jointly modeled entity coreference and event coreference. They first obtained the predicate-argument structures through a SRL system, and then heuristically found the relation between entities and events by matching the semantic roles with entity annotations and event annotations. After that, they encoded the embedding of related entities (events) into the event (entity) embedding. Finally, the pairwise entity coreference scorer and event coreference scorer were trained alternately and iteratively based on these embeddings.

Based on the Barhom's method (2019), Meged et al. (2020) introduced Chirps resources (Shwartz et al., 2017) in a distant supervision manner to enhance event coreference resolution. Chirps is a project focusing on predicate (verbal event mention[1]) paraphrase. It collects news headlines on tweets, annotates the sentences with semantic roles, and then evaluates the confidence of predicate paraphrase through heuristic methods. Meged et al. first re-ranked the paraphrased predicates in Chirps by using the event coreference annotations in ECB+ corpus (Cybulska and Vossen, 2014) as supervision signals. Then, Chirps provides predicate paraphrase pairs to the joint method to improve event coreference resolution.

Differing from the method proposed by Meged et al. (2020), our approach models the event-specific paraphrases between sentences and can capture the deep relations between events.

3 The EPASE Model

Given a document collection $\mathcal{D}$, where each document d_i consists of a series of sentences $\{s_0, s_1, ...s_n\}$. In each sentence, there may be zero or more events. The event coreference resolution task is to find all the events that refer to the same real-world event with the given event in the document set. In our EPASE model, we formulate it as a pairwise similarity identification task. The overall architecture of our model is shown in Figure 1. We first conduct document clustering to narrow the search scope and construct pairwise samples for model training by sampling. Then we employ the event paraphrase identification module to model the event-specific paraphrase similarity vector $\mathbf{S}_{para}$ and the semantic similarity evaluation module to model the semantic similarity vector $\mathbf{S}_{sem}$ between events. The final similarity vector $\mathbf{S}$ is calculated by $\mathbf{S}_{sem}$ and $\mathbf{S}_{para}$.

3.1 Data preparation

For data preparation, we first conduct document clustering to narrow the search scope and construct pairwise samples for model training by sampling

3.1.1 Document clustering

Our model starts with the preprocessing step of document clustering, which clusters the input documents $\mathcal{D}$ into a set of document clusters $\mathcal{C}$. It has proved to be a very effective means to reduce the search space and mitigate errors (Lee et al., 2012; Barhom et al., 2019), especially when we need to handle both within- and cross-document event coreference.

Following Barhom et al. (2019), we convert the documents into tf-idf vectors after removing stop words and retaining the unigrams, bigrams, and trigrams. Then, we adopt the K-Means algorithm to cluster the documents into different clusters. To improve efficiency and effectiveness, in the subsequent procedures, we only identify the coreference link between events that co-occur in the same document cluster.

[1]Note that, event mentions can be verbs, nouns, pronouns and even adjectives.

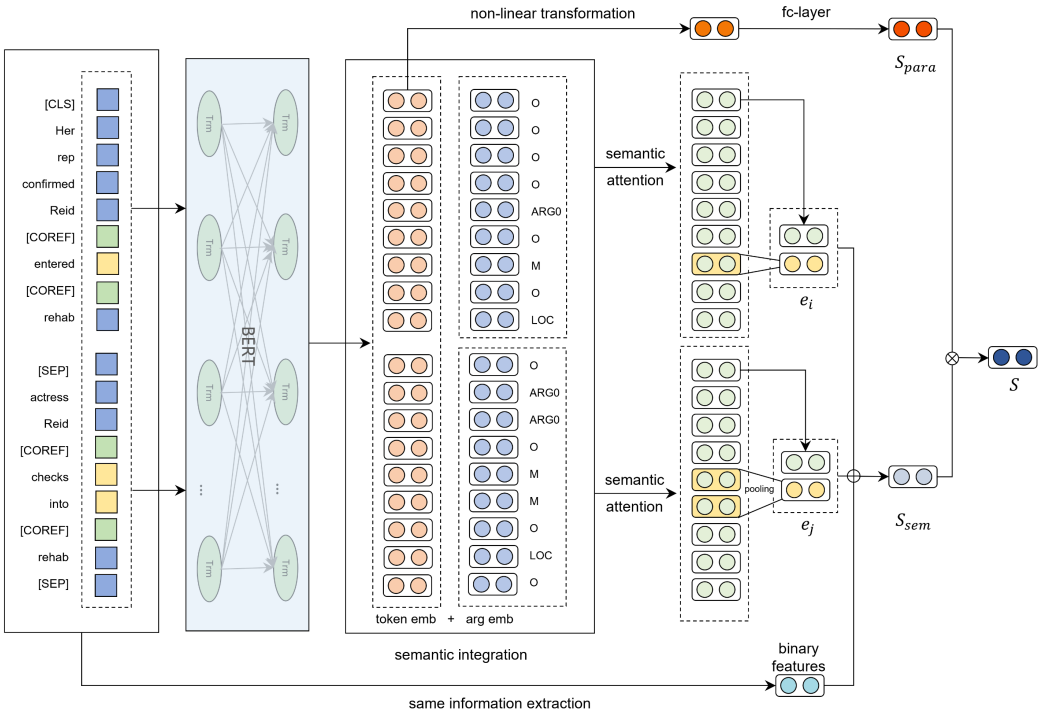


Figure 1: The overall architecture of the EPASE model.

3.1.2 Mention-pair sampling

After obtaining the document clusters, we pair the event mentions in the same document clusters to construct the mention-pair set. For each document cluster $\mathcal{C}_p \in \mathcal{C}$, all event mentions appearing in it constitute the mention set $\mathcal{E}_p$, then the mention-pair set can be expressed as $\mathcal{MP} = \{\langle m_{ip}, m_{jp}, r\rangle | m_{ip}, m_{jp} \in \mathcal{E}_p; i \neq j\}$, where r represents whether m_{ip} and m_{jp} are coreferential event mentions.

It's easy to find that the coreference relations between event mentions are sparse, and the sparsity is more serious when the number of documents in the document cluster increases. Thus, we apply sampling strategies to alleviate it. For positive samples in $\mathcal{MP}$, we set an oversampling rate $\lambda_o(\lambda_o \geq 1)$, and for negative samples, we conduct a downsampling procedure with sampling probability $\lambda_d(\lambda_d \leq 1)$. The sampling strategies are only conducted on the training data.

The final data pair set $\mathcal{DP}$ consists of a series of sample pairs. For each sample $\langle m_i, s_i, m_j, s_j, r\rangle$, $\langle m_i, m_j, r\rangle \in \mathcal{MP}$ and s_i, s_j are the corresponding sentences containing m_i, m_j respectively.

3.2 Event-specific paraphrase identification

An event consists of one event mention and some event arguments, and the coreferential events are different statements for it. Currently, many methods (Lee et al., 2012; Cybulska and Vossen, 2015b) construct features through complex data preprocessing steps to capture the paraphrase features as much as possible, such as event argument matching, event mention matching, and event mention lemma matching. These steps can intuitively find some common information between events. However, they are very time-consuming and requires the careful artificial construction of the corresponding features. Also, it is easy to ignore the context information by simply using the matching rules, resulting in mis-coreference of events with the same event mentions or event arguments.

Therefore, a better way is to keep the event mention and event arguments in the sentences and conduct paraphrase identification between sentences. However, since there are often multiple events in a sentence, it is difficult to identify the paraphrase relation without distinguishing which event is the focused one. Here, we try to capture the event-specific paraphrase relation between sentences, which focuses on only the event-specific context. That is, when we focus on the specific event or analyze the sentence from the perspective of a specific event, other events or semantic components can be ignored.

To this end, we need to force the model to pay attention to specific events in the sentence pair. Here, we add a special token before and after the event mentions to emphasize the events, and then use the attention mechanism to gather the information of different semantic components in the sentence.

Detailedly, for each sample $\langle m_i, s_i, m_j, s_j, r\rangle$, we use BERT (Devlin et al., 2019) as a sentence-pair (s_i, s_j) encoder to capture paraphrase feature information, and gather important semantic structure information through the self-attention mechanism. To be in line with BERT , we add [CLS] and [SEP] to connect two sentences. To force the model to pay attention to the event-specific semantic structure, we adopt a straightforward strategy that encapsulates the event mentions in sentences with the special token [COREF][2]. For example, the event pair (*strike*, *hits*) and the sentence pair (*"A powerful quake strikes the Indonesian province of Aceh"*, *"Dozens injured as the earthquake hits Aceh"*) will be transformed into:

- [CLS] *A powerful quake* [COREF] *strikes* [COREF] *the Indonesian province of Aceh* [SEP] *Dozens injured as the earthquake* [COREF] *hits* [COREF] *Aceh* [SEP]

Suppose the input sequence T consists of a series of tokens, where $T = \{t_0, t_1, ...t_n\}$. Following Devlin et al. (2019), we use the token embedding of [CLS], i.e., t_0 as the representation of the sentence pair. Then we pass it through a fully connected layer, followed by a non-linear activation operation, and another fully-connected layer to obtain the binary event-specific paraphrase vector $\mathbf{S}_{para}(m_i, m_j)$ of m_i and m_j, i.e.,

$$\mathbf{S}_{para}(m_i, m_j) = \mathbf{W}_{p1}\left[\tanh\left(\mathbf{W}_{p0} \cdot \mathcal{B}(t_0) + b_{p0}\right)\right] + b_{p1}, \tag{1}$$

where $\mathcal{B}$ is the BERT embedding; $\mathbf{W}_{p0}$ and $\mathbf{W}_{p1}$ are the weight matrix of the two fully-connected layers, respectively; b_{p0} and b_{p1} are the biases of these two layers.

3.3 Semantic similarity evaluation

It's hard for the paraphrase identification module to recognize all event arguments without semantic information. Therefore, we also explicitly introduce argument role information into the model to provide a complement to the paraphrase feature, especially when the event arguments in the sentence is far from the event mentions or does not match exactly between event pairs.

Specifically, we first carry out SRL on the input two sentences. Then, we combine the token embedding obtained from the BERT encoder with argument label embedding and obtain a joint embedding through a semantic integration module. At last, we conduct pooling strategies to get the event embedding e_i, e_j, and calculate the event semantic similarity based on them.

3.3.1 Semantic role labeling

We identify semantic roles of events using a SRL system. The output result for each predicate is a label sequence which consists of semantic role labels of each token. In this paper, we use the same semantic role annotations as the PropBank (Palmer et al., 2005). In addition, we add two labels O and M, O stands for the empty role token and M stands for the event mention token. Usually, there are multiple events in a sentence, so there will be multiple semantic role sequences. For each event, we choose the sequence in which the predicate matches the event mentions.

Previous methods (Lee et al., 2012; Choubey and Huang, 2017; Barhom et al., 2019) usually match the output of the SRL system with the annotated entities in the dataset to obtain four kinds of event arguments, i.e., ARG0 (agent), ARG1 (patient), ARGM-TIM (time), ARGM-LOC (location). This not only requires additional entity annotations, but also limits the realization of an end-to-end event coreference resolution system. Besides, the matching method often introduces errors due to complex sentence structures, resulting in error propagation. Therefore, we remove the matching process and regard the semantic role labels as embeddings by using a lookup table to map these labels to vectors $R = \{\mathbf{r}_0, \mathbf{r}_1, ..., \mathbf{r}_m\}$.

Since current SRL systems can only handle verb predicates, event mentions such as nouns and pronouns cannot be handled well. Lee et al. (2012) used a heuristic means that considering the possessor of

[2]For discontinuous event mentions, we add [COREF] before the first word and after the last word.

a nominal event as its ARG0 (e.g., *the AMD's deal*) to cover nominal events. But we find it will introduce errors and noise for model training. Therefore, for events that are not covered[3] by SRL system, we leave all its argument roles empty. In future work, we will train a SRL model that can handle nouns, pronouns and adjectives to handle this situation.

3.3.2 Semantic integration

For the two sentences s_i and s_j in each sample, we add a [CLS] token at the beginning of each sentence to aggregate the overall sentence information. Each sentence consists of a series of vectors $s = \{\mathbf{v}_0, \mathbf{v}_1, ..., \mathbf{v}_n\}$, where $\mathbf{v}_k = [\mathcal{B}(t_k)||\mathbf{r}_k]$ $(k = 0, 1, \ldots, n)$ is the concatenation of token embedding from BERT encoder and semantic role embedding. We use a multi-layer multi-head transformer encoder (Vaswani et al., 2017) to encode the semantic information into the new token embedding $\mathbf{z}$, i.e.,

$$\mathbf{h}_k^l = \mathrm{Transformer}_l(\mathbf{v}_k) \quad (l = 1, 2, \ldots, n), \tag{2}$$

$$\mathbf{z}_k = \mathbf{W}_h \cdot [\mathbf{h}_k^1||\mathbf{h}_k^2...||\mathbf{h}_k^n], \tag{3}$$

where $[\cdot||\cdot]$ represents the concatenation operation, $\mathrm{Transformer}_l$ denotes the l-th head of the transformer encoder, $\mathbf{W}_h$ is the weight matrix of head concatenation operation and n is the number of heads.

Then, the embedding of [CLS], i.e., $\mathbf{z}_0$, is adopted as the sentence embedding $\mathbf{g}$. And we employ average pooling to obtain the event mention embedding $\mathbf{x}$ and the event embedding $\mathbf{e}$:

$$\mathbf{x} = \frac{1}{q - p + 1} \sum_{k=p}^{q} \mathbf{z}_k, \tag{4}$$

$$\mathbf{e}_u = \tanh(\mathbf{W}_u \cdot [\mathbf{g}_u||\mathbf{x}_u] + b_u) \quad u \in i, j, \tag{5}$$

where p, q are the indices of the first token and last token; $\mathbf{W}_u$ and b_u are the weight matrix and the bias term, respectively.

In addition, we enrich our model with some pairwise binary features $\Phi(m_i, m_j)$, indicating whether the two event mentions have the same head lemma word and the same event type[4].

Based on them, we can get the binary semantic similarity vector $\mathbf{S}_{sem}(m_i, m_j)$ of events m_i and m_j:

$$\mathbf{S}_{sem}(m_i, m_j) = \mathbf{W}_{sem} \cdot [\mathbf{e}_i||\mathbf{e}_j||\mathbf{e}_i \circ \mathbf{e}_j||\Phi(m_i, m_j)] + b_{sem}, \tag{6}$$

where $\circ$ is the element-wise multiplication operation; $\mathbf{W}_{sem}$ and b_{sem} are the weight matrix and bias, respectively.

3.4 Training and inference

Based on paraphrase similarity vector $\mathbf{S}_{para}$ and semantic similarity vector $\mathbf{S}_{sem}$ of m_i and m_j, the final binary similarity vector $\mathbf{S}$ and the coreference probability y_{ij} are computed as follows:

$$\mathbf{S}(m_i, m_j) = \mathbf{W}_S \cdot [\mathbf{S}_{para}(m_i, m_j)||\mathbf{S}_{sem}(m_i, m_j)] + b_S, \tag{7}$$

$$y_{ij} = \mathrm{softmax}(\mathbf{S}(m_i, m_j)[1]) = \frac{exp(\mathbf{S}(m_i, m_j)[1])}{\sum_{k\in\{0,1\}} exp(\mathbf{S}(m_i, m_j)[k])}, \tag{8}$$

where $\mathbf{W}_S$ and b_S are the corresponding weight matrix and bias, the $\mathbf{u}[k]$ denotes the k-th dimension of the vector $\mathbf{u}$.

During training, we apply dropout before the last fully-connected layer of obtaining $\mathbf{S}_{sem}$, $\mathbf{S}_{para}$ and set the same dropout ratio in the transformer encoder layers. The training object is to minimize the binary cross-entropy loss L:

$$L = -\frac{1}{N} \sum_{n=1}^{N} \left[y^n log\hat{y}^n + (1 - y^n)\, log\,(1 - \hat{y}^n) \right]. \tag{9}$$

During inference, m_i, m_j will be considered as coreferential events only when both the coreference probabilities y_{ij} and y_{ji} are greater than the score threshold δ.

[3] According to our statistics, the results of SRL system can cover about 60% of event mentions.

[4] The head lemma word can be obtained from lemmatization tools and the event type can be obtained from the dataset.

4 Experiments

4.1 Experimental setup

Our experiments are conducted on the ECB+ corpus (Cybulska and Vossen, 2014), which is the largest dataset containing WD and CD event coreference annotations. It consists of documents accumulated from Google News. These documents are clustered into different topics according to the seminal events. In the ECB+ corpus, the salient events and entities are annotated, but the roles of the entities in the corresponding events are not annotated.

Following the setup of Cybulska and Vossen (2015a) and Barhom et al. (2019), we use a subset of the annotations that has been manually reviewed and checked for correctness. In this setting, singleton events (events that have no coreferential events) are included, and the 43 topics in the ECB+ corpus are divided into a training set, a dev set, and a test set. Table 1 presents the statistics of the data.

	Train	**Dev**	**Test**	**Total**
# Topics	25	8	10	43
# Documents	574	196	206	976
# Sentences	1037	346	457	1840
# Event mentions	3808	1245	1780	6833
# Event chains	1527	409	805	2741
# Avg. chain length	2.49	3.04	2.21	2.49
# Avg. WD chain length	1.23	1.26	1.27	1.24
# Avg. CD chain length	2.44	2.99	2.17	2.44

Table 1: The statistics of ECB+ corpus.

We evaluate the event coreference performance on four widely used coreference resolution metrics: **MUC** (Vilain et al., 1995), $\mathbf{B}^3$ (Bagga and Baldwin, 1998), $\mathbf{CEAF}_e$ (Luo, 2005), and **CoNLL F1** (Pradhan et al., 2014), which is the average of **MUC**, $\mathbf{B}^3$ and $\mathbf{CEAF}_e$.

4.2 Baselines

We adopt three categories of baselines, i.e., feature or template based methods: **Cluster+Lemma** and **CV** (Cybulska and Vossen, 2015b); neural network based methods: **KCP** (Kenyon-Dean et al., 2018) and **Cluster+KCP** (Barhom et al., 2019); information enhanced ones: **Joint** (Barhom et al., 2019) and **Joint+Chirps** (Meged et al., 2020). They are all representative or state-of-the-art. Among them, Cluster+Lemma and Cluster+KCP are variants of the lemma rule (Lee et al., 2012) and KCP, respectively:

Cluster+Lemma: a heuristic method that includes document clustering and lemma matching. We first cluster the documents, using the method stated in Section 3.1, and then group event mentions in the same document cluster according to whether they have the same head lemma.

Cluster+KCP: a model that adds the document clustering module to the KCP baseline, proposed by Barhom et al. (2019). We make comparison with this model to distinguish the effect of document clustering from the other parts of our model.

Besides, we add another baseline $\mathbf{BERT}_{para}$, which use BERT to identify common sentence paraphrases. We compare our model with it to reflect the necessity of being event-specific while conducting paraphrase identification.

Since Yang et al. (2015) and Choubey and Huang (2017) adopted a different experimental setup, which had been criticized by Barhom et al. (2019) for using the full corpus with known annotation errors and ignoring singleton events, we do not apply them as baselines following Barhom et al. (2019).

4.3 Implement details

In our experiments, we use AllenNLP (Gardner et al., 2018) to conduct SRL on input sentences, use spaCy (Honnibal and Montani, 2017) to obtain the head lemma of event mentions. For document clustering, we use K-Means algorithm, and set K to 20 following Barhom et al. (2019). For data sampling,

we set λ_o to 1, and the downsampling probability λ_d to 0.5. For the hyperparameters in our model, we set the dimension of token embedding to 768 (the same dimension size as BERT-base (Devlin et al., 2019)), the dimension of semantic role embedding to 384 (half of the dimension size of token embedding), and the total dimension of binary features to 50. For the semantic integration module, we use a two-layer multi-head transformer, set the number of attention heads to 4, the hidden layer dimension of the feed-forward network to 2048, and use GeLU (Hendrycks and Gimpel, 2016) as the activation function. During training, we set the learning rate to 2e-5, the dropout ratio to 0.1, the maximum sentence pair length to 200, and use the AdamW (Loshchilov and Hutter, 2017) optimizer with a minibatch size of 24. During inference, the threshold δ is set to 0.5.

4.4 Experimental results

Model	MUC			B^3			$CEAF_e$			CoNLL
	P	R	F1	P	R	F1	P	R	F1	Avg F1
$BERT_{para}$	53.4	23.7	32.8	81.9	51.0	62.8	46.9	78.4	58.7	51.4
Cluster+Lemma	79.9	76.5	78.1	85.0	71.7	77.8	71.7	75.5	73.6	76.5
CV	75	71	73	78	71	74	-	-	64	73
KCP	71	67	69	67	71	69	67	71	69	69
Cluster+KCP	79.3	68.4	73.4	**87.2**	67.2	75.9	66.4	77.4	71.5	73.6
Joint	84.5	77.6	80.9	85.1	76.1	80.3	73.8	81.0	77.3	79.5
Joint+Chirps†	84.7	78.7	81.6	85.9	75.9	80.5	74.8	**81.1**	77.8	80.0
EPASE	**85.6**	**89.3**	**87.5**	77.6	**89.7**	**83.2**	**84.5**	80.1	**82.3**	**84.3**

Table 2: Within- and cross-document event coreference results on the ECB+ test set († indicates that the method uses external data).

Table 2 presents the performance of the models for within- and cross-document event coreference. Overall, our model shows a consistent improvement and achieves the best results on all the metrics. Compared with the state-of-the-art Joint+Chirps baseline, our EPASE model achieves a 4.3% improvement on CoNLL F1, even though Joint+Chirps additionally uses entity annotation information and the data resource from Chirps (Shwartz et al., 2017). The most significant improvement of our model comes from the recall of MUC and B^3, which increases the scores by 10.6% and 13.8%, respectively.

In addition, we also conduct experiments on a special part of the data samples (within-document) (See Table 3). We don't need document clustering when evaluating models on within-document coreference resolution, so the Cluster+KCP is equal to the KCP baseline.

In Table 3, the MUC scores of all models are significantly lower than those of B^3 and $CEAF_e$. This is because when the cross-document coreference links are cut off, most of the events become singleton event (as shown in Table 1, the average length of WD coreference chains is 1.24). B^3 and $CEAF_e$ will be greatly influenced by these singleton nodes. In this case, MUC is more suitable for comparing the performances between models. We can observe that our model achieves an improvement of 8.3% on MUC F1 as well as a 4.2% improvement on CoNLL F1 when compared with the Joint baseline.

Model	MUC			B^3			$CEAF_e$			CoNLL
	P	R	F1	P	R	F1	P	R	F1	Avg F1
$BERT_{para}$	18.6	15.4	16.8	84.5	81.6	83.1	77.9	81.6	79.7	59.9
Cluster+Lemma	78.8	50.4	61.5	**96.8**	88.4	92.4	84.7	92.9	88.6	80.8
KCP	57	69	63	90	**94**	92	90	86	88	81
Joint	74	65.8	69.7	94.2	91.5	92.8	88.5	91.2	89.8	84.1
EPASE	**82.1**	**74.3**	**78.0**	95.6	93.8	**94.7**	**91.1**	**93.4**	**92.3**	**88.3**

Table 3: Within-document event coreference results on the ECB+ test set.

These results reconfirm that, compared to using discrete manual features and argument slots with a fixed number to obtain event embedding, introducing paraphrase features and integrating argument label embeddings into event embeddings will greatly improve the performance of event coreference resolution.

To further study the necessity of each component of the proposed model, we ablate a component from the full model to validate its contributions each time. The results are demonstrated in Table 4. -S_{para}, -S_{sem} and -bin refer to the removal of the paraphrase similarity component, the removal of the semantic similarity component, and the removal the pairwise binary features. We can observe that all components contribute to the performance improvement, among which both the paraphrase similarity component and the semantic similarity component play a great role in improving the experimental results. It is worth mentioning that when only S_{sem} is used, our method still surpasses the existing SOTA model by 2.7% on CoNLL F1, which shows that the method of incorporating the argument role embedding with word embedding, and then obtaining the event expression through the attention mechanism can get better event embedding, thus lead to better coreference results.

Model	**MUC**	**B^3**	**CEAF$_e$**	**CoNLL**	Δ
All	87.5	83.2	82.3	84.3	
-S_{sem}	86.3	80.8	79.1	82.1	-2.2
-S_{para}	86.7	81.1	80.4	82.7	-1.6
-bin	87.4	83.0	81.9	84.1	-0.2

Table 4: Ablation study for event coreference.

4.5 Analysis of event-specific paraphrase

To figure out whether the EPASE model actually models the event-specific paraphrase features, we select a typical sample from the test set and illustrate the heat map of the attention weights of the tokens in this event pair in Figure 2.

From this figure, we can observe that after adding [COREF] to emphasize the event mentions, the model will use the [COREF] tokens to gather event information in another sentence, including the mention and arguments of another event. Meanwhile, little attention will be paid to the irrelevant events (the *injured* event in this example).

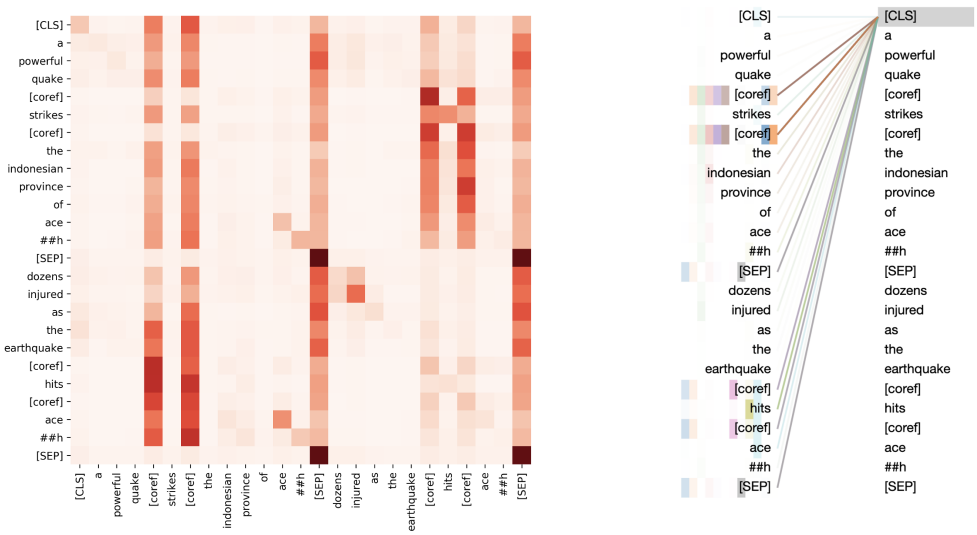
We also draw a chart Figure 2(b) to display how event pairs information is aggregated to the [CLS] token, which is used as event-specific paraphrase embedding to calculate the paraphrase similarity. The thickness of the line reflects the attention score. We can find that most of the information flowing to [CLS] comes from [COREF], context, and event arguments. And we notice that it's hard for the paraphrase component to capture all arguments (*quake* and *earthquake* in this sample) without introducing more semantic information.

Therefore, the proposed model EPASE captures the event-specific paraphrase features and is thus of high generalization ability.

5 Conclusions and future work

We proposed EPASE to resolve event coreference, which integrates event-specific paraphrases and argument-aware semantic embeddings. Compared with obtaining event embeddings within single sentences and calculating event similarity based on them, EPASE identifies event-specific paraphrases between sentences to capture the correlation between two events, which is more in line with human cognition. Meanwhile, to make up for the situation that not all the event arguments can be recognized in the paraphrase identification, we further introduce the embeddings of argument roles to obtain argument-aware semantic embedding of events. Thus, EPASE is of high generalization ability and scalability. Experimental results manifest the remarkable superiority of EPASE.

In the future, we will strengthen our SRL component to handle event mentions which are pronouns, nouns, and adjectives. Also, we will further reduce the reliance on the event mention annotations and



(a) (b)

Figure 2: Visualization of attention scores between tokens

construct an end-to-end event coreference resolution framework.

Acknowledgements

We are grateful to Long Bai, Yunqi Qiu and Zixuan Li for valuable discussion. We also appreciate anonymous reviewers for constructive review comments.

This work is supported by the National Key Research and Development Program of China under grants No. 2016YFB1000902, etc., the National Natural Science Foundation of China under grants Nos. U1911401, 61772501, U1836206, 91646120, and 61722211, the GF Innovative Research Program, the Beijing Academy of Artificial Intelligence (BAAI) under grant No. BAAI2019ZD0306, and the Lenovo-CAS Joint Lab Youth Scientist Project.

References

Amit Bagga and Breck Baldwin. 1998. Algorithms for scoring coreference chains. In *The first international conference on language resources and evaluation workshop on linguistics coreference*, volume 1, pages 563–566. Citeseer.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4179–4189.

Cosmin Adrian Bejan and Sanda Harabagiu. 2010. Unsupervised event coreference resolution with rich linguistic features. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1412–1422.

Prafulla Kumar Choubey and Ruihong Huang. 2017. Event coreference resolution by iteratively unfolding inter-dependencies among events. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2124–2133.

Agata Cybulska and Piek Vossen. 2014. Using a sledgehammer to crack a nut? lexical diversity and event coreference resolution. In *LREC*, pages 4545–4552.

Agata Cybulska and Piek Vossen. 2015a. Translating granularity of event slots into features for event coreference resolution. In *Proceedings of the the 3rd Workshop on EVENTS: Definition, Detection, Coreference, and Representation*, pages 1–10.

Agata Cybulska and Piek Vossen. 2015b. "bag of events" approach to event coreference resolution. supervised classification of event templates. *IJCLA*, page 11.

Marie-Catherine De Marneffe, Anna N Rafferty, and Christopher D Manning. 2008. Finding contradictions in text. In *Proceedings of ACL-08: HLT*, pages 1039–1047.

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

Elisa Ferracane, Iain Marshall, Byron C Wallace, and Katrin Erk. 2016. Leveraging coreference to identify arms in medical abstracts: An experimental study. In *Proceedings of the Seventh International Workshop on Health Text Mining and Information Analysis*, pages 86–95.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F Liu, Matthew Peters, Michael Schmitz, and Luke Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Matthew Honnibal and Ines Montani. 2017. spacy 2: Natural language understanding with bloom embeddings. *convolutional neural networks and incremental parsing*, 7(1).

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. *NAACL HLT 2018*, page 1.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

Heeyoung Lee, Marta Recasens, Angel Chang, Mihai Surdeanu, and Dan Jurafsky. 2012. Joint entity and event coreference resolution across documents. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 489–500. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Xiaoqiang Luo. 2005. On coreference resolution performance metrics. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 25–32.

Yehudit Meged, Avi Caciularu, Vered Shwartz, and Ido Dagan. 2020. Paraphrasing vs coreferring: Two sides of the same coin. *arXiv preprint arXiv:2004.14979*.

Martha Palmer, Daniel Gildea, and Paul Kingsbury. 2005. The proposition bank: An annotated corpus of semantic roles. *Computational linguistics*, 31(1):71–106.

Sameer Pradhan, Xiaoqiang Luo, Marta Recasens, Eduard Hovy, Vincent Ng, and Michael Strube. 2014. Scoring coreference partitions of predicted mentions: A reference implementation. In *Proceedings of the conference. Association for Computational Linguistics. Meeting*, volume 2014, page 30. NIH Public Access.

Vered Shwartz, Gabriel Stanovsky, and Ido Dagan. 2017. Acquiring predicate paraphrases from news tweets. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (* SEM 2017)*, pages 155–160.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*.

Johannes Welbl, Pontus Stenetorp, and Sebastian Riedel. 2018. Constructing datasets for multi-hop reading comprehension across documents. *Transactions of the Association for Computational Linguistics*, 6:287–302.

Bishan Yang, Claire Cardie, and Peter Frazier. 2015. A hierarchical distance-dependent bayesian model for event coreference resolution. *Transactions of the Association for Computational Linguistics*, 3:517–528.