Improving Event Coreference Resolution Using Document-level and Topic-level Information

Sheng Xu, Peifeng Li and **Qiaoming Zhu**
School of Computer Science and Technology
Soochow University
sxu@stu.suda.edu.cn, {pfli, qmzhu}@suda.edu.cn

Abstract

Event coreference resolution (ECR) aims to cluster event mentions that refer to the same real-world events. Deep learning methods have achieved SOTA results on the ECR task. However, due to the encoding length limitation, previous methods either adopt classical pairwise models based on sentence-level context or split each document into multiple chunks and encode them separately. They failed to capture the interactions and contextual cues among those long-distance event mentions. Besides, high-level information, such as event topics, is rarely considered to enhance representation learning for ECR. To address the above two issues, we first apply a Longformer-based encoder to obtain the document-level embeddings and an encoder with a trigger-mask mechanism to learn sentence-level embeddings based on local context. In addition, we propose an event topic generator to infer the latent topic-level representations. Finally, using the above event embeddings, we employ a multiple tensor matching method to capture their interactions at the document, sentence, and topic levels. Experimental results on the KBP 2017 dataset show that our model[1] outperforms the SOTA baselines.

1 Introduction

Within-document Event Coreference Resolution (ECR) is the task of grouping the event mentions (i.e., triggers) that occur in a document into clusters such that each cluster represents a unique real-world event. ECR is essential for information aggregation and can help many downstream applications, such as abstractive summarization (Li et al., 2016), central event detection (Choubey et al., 2018), and discourse parsing (Lee et al., 2020).

Although there studies employ clustering methods (Chen and Ji, 2009; Phung et al., 2021), most work converts ECR into an event-pair classification task (Krause et al., 2016; Huang et al., 2019; Lu and Ng, 2021a), i.e., judging whether two event mentions are coreferent. Therefore, event encoding becomes an essential step in ECR. Earlier work focuses on extracting hand-crafted features (Chen et al., 2009; Cybulska and Vossen, 2015; Lu and Ng, 2018), while recent studies encode triggers (i.e., the words that most clearly describe the event), contextual linguistic cues (Nguyen et al., 2016; Huang et al., 2019; Lai et al., 2021), and identified arguments (i.e., the participants of a specific event) (Zeng et al., 2020) using various neural models. Recently, benefiting from the development of the Transformer-based models, event encoding (Krause et al., 2016; Huang et al., 2019; Lu et al., 2022; Lu and Ng, 2021a) has also changed from using CNN/RNN to BERT, SpanBERT, obtaining better event representations. However, at least two limitations exist in the above studies.

First, due to the limitation of encoding length, current studies either adopt variant classical pairwise models based on sentence-level context or split a document into chunks and process them separately to learn embeddings based on segment-level context. This way, the ECR models actually predict coreferences based only on the local information. Obviously, incorporating document-level contextual cues among event mentions is important for ECR, especially for those long-distance event mention pairs, considering their relations are hard to judge only based on local information. To address the above issue, Tran et al. (2021) implement document structures to capture cross-segment interactions; however, the graph construction relies on predicted entities, entity coreference chains, and dependency trees, which inevitably bring the noise to the model. In this paper, we first introduce a Longformer-based (Beltagy et al., 2020) encoder to learn representations based on full document-level context. Then, we apply a BERT-based encoder with a trigger-mask mechanism to mine detailed

[1] Code is available at https://github.com/jsksxs360/event-coref-emnlp2022

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 6765–6775
December 7-11, 2022 ©2022 Association for Computational Linguistics

cues from sentence-level contexts as a beneficial complement. To our knowledge, this is the first to apply the full-document encoding to ECR.

Second, most existing ECR methods only use representations obtained by text encoder while ignoring high-level information, such as event topics. Since events are usually mentioned repeatedly only when elaborating new aspects or further information (Choubey and Huang, 2018), it is challenging to identify coreferent events based only on sentence-level or document-level information, especially for those main events (Choubey et al., 2020), which advance the content and form long event chains. Previous studies (Choubey and Huang, 2018; Lu and Ng, 2020) try incorporating event topic information to improve ECR; however, they use integer linear programming or traditional machine learning models. It is difficult to combine these topic models with deep learning methods directly. To this end, we propose a neural event topic model to infer the latent topic distribution for every event mention. In this way, our ECR model can utilize topic-level associations and interactions to predict coreference chains. Finally, we feed these different level representations into a scorer to judge coreferences for all event mention pairs by applying multiple tensor matching methods. The experimental results on KBP 2017 dataset demonstrate the benefits of our proposed model. We summarize the contributions of our work as follows.

- Our proposed ECR model is the first to process the entire document at once and apply the full-document event encoding using a Longfomer-based encoder;
- Our proposed ECR model introduces a neural event topic model to infer the latent topic-level event representations;
- Our proposed ECR model is simple but effective and requires no additional information.

2 Related Work

Coreferent events normally have the same type of triggers and entity-coreferent arguments (i.e., event participants). Therefore, resolving event mentions has been considered more challenging than entity coreference resolution due to the more complex event structures (Yang et al., 2015). Most of the current work for ECR focuses on event mention representation learning (Huang et al., 2019; Zeng et al., 2020; Tran et al., 2021).

Our work focuses on the within-document ECR, where input event mentions appear in the same documents; however, we also note previous studies on cross-document ECR (Lee et al., 2012; Choubey and Huang, 2017; Kenyon-Dean et al., 2018; Barhom et al., 2019), which usually require clustering documents first and then resolving coreferent events mentioned in each document cluster. As such, for within-document ECR, previous methods have applied various machine learning models with hand-engineered features (Chen et al., 2009; Cybulska and Vossen, 2015; Peng et al., 2016; Lu and Ng, 2018), spectral graph clustering (Chen and Ji, 2009), joint inferencing using Integer Linear Programming or Markov Logic Networks (Chen and Ng, 2016; Lu et al., 2016), joint modeling with multiple related tasks (Lu and Ng, 2017, 2021a,b; Kriman and Ji, 2021), and typical pipeline approaches (Liu et al., 2014; Peng et al., 2016; Krause et al., 2016; Tran et al., 2021). In particular, Lai et al. (2021) introduce gate mechanisms to mitigate error propagation by controlling the information flows from the input symbolic features. In addition, some studies explore the relevance of event coreferences and discourse structures, e.g., topic structures and content types (Choubey and Huang, 2018; Choubey et al., 2020; Lu and Ng, 2020).

However, due to the limitation of encoding length, current deep learning methods either adopt classical pairwise models (Krause et al., 2016; Huang et al., 2019; Zeng et al., 2020) or split each document into multiple chunks and encode them separately (Lu and Ng, 2021a,b; Tran et al., 2021). These methods learn event mention representations based only on the local context, making ECR models difficult to determine coreferences between long-distance event mention pairs. In particular, to alleviate the encoding separation caused by splitting documents, Tran et al. (2021) construct graph-form document structures to capture the interactions between distant sentences. Compared with previous deep learning work for ECR, we present a novel representation learning framework based on both document-level and sentence-level context and apply a neural event topic model to further provide topic-level representations.

3 Model

Formally, given an input document $D = \{t_1, t_2, ..., t_N\}$ (of N tokens) and a set of event mentions $E = \{e_1, e_2, ..., e_{|E|}\}$ in D, ECR seeks

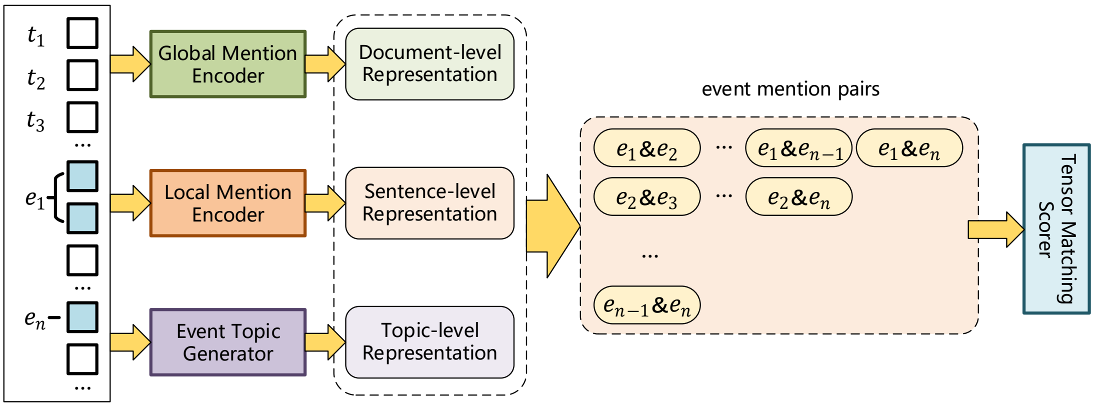

Figure 1: The overall framework of our ECR model.

to group the event mentions in E into clusters. As such, all event mentions in each cluster are coreferent. Figure 1 presents an overview of our ECR model, which consists of four main components: (i) Global Mention Encoder (GME) to learn document-level event representations, (ii) Local Mention Encoder (LME) to obtain sentence-level event representations, (iii) Event Topic Generator (ETG) to infer latent topic-level event representations, and (iv) Tensor Matching Scorer (TMS) to predict the probability that two event mentions are coreferent.

3.1 Global Mention Encoder

Using the long-sequence encoding of Longformer (Beltagy et al., 2020), we can process the entire document at once instead of splitting each document into multiple chunks like in the previous work. With the help of sliding window attention, even long-distance event mentions can learn full-context-based representations without losing the semantic interactions between each other and requiring inputting additional global structures. Specifically, we first convert each token $t_i \in D$ into semantic embeddings by feeding D into a pre-trained Longformer model. Since the event mentions e_i may be phrases containing multiple words, and the tokenization scheme may split one word into multiple tokens, we apply the attention mechanism on top of the hidden vectors $\boldsymbol{h}_j$ of the j-th tokens in D in the last layer of Longformer model to obtain the representation vector $\boldsymbol{g}_i$ for e_i as follows.

$$\boldsymbol{g}_i = \sum_{j=p}^{q} \alpha_j \boldsymbol{h}_j \quad (1)$$

$$\alpha_i = \frac{\exp(w_i)}{\sum_{j=p}^{q} \exp(w_j)} \qquad w_i = \boldsymbol{w}_g^\top \boldsymbol{h}_i \quad (2)$$

where p and q are the positions of the start and end token of the event mention e_i, respectively, and $\boldsymbol{w}_g$ is the model parameter. Since all hidden vectors $\boldsymbol{h}_j$ are learned based on full document context, we treat $\boldsymbol{g}_i$ as the document-level representation for each event mention e_i.

3.2 Local Mention Encoder

Due to the complex structure of events, it is also important to mine detailed cues from local contexts where the event triggers are located. Some studies perform tensor matching on event templates filled with the recognized arguments (Choubey and Huang, 2017; Barhom et al., 2019), and others (Zeng et al., 2020) directly use semantic role labels as embeddings to alleviate the cascading errors. However, all of these methods rely on the prediction of syntactic or semantic parsing systems, such as semantic role labeling, bringing additional noises. Inspired by the studies on event extraction (Tong et al., 2020; Liu et al., 2020a) that utilize the Masked Language Model (MLM) to mine event knowledge from context, we propose a local mention encoder with a trigger-mask mechanism to learn sentence-level event representations. This can remedy the weaknesses of document-level event representations in capturing local clues.

Specifically, for each event mention, we replace its event trigger with placeholders, i.e., `[MASK]` tokens, and require the model to predict the corresponding event subtype. As such, the model is compelled to implicitly mine event clues from the contexts surrounding the triggers to make the predictions. Formally, given the local context of each event mention (the sentence hosting the event trigger) $S = \{t_1, t_2, ..., t_M\}$ (of M tokens) and the start and end token corresponding to the event trigger, t_p and t_q, the cloze-style input to the local encoder is as follows.

$$S' = t_1, ..., t_{p-1}, \texttt{[MASK]}, ..., \texttt{[MASK]}, t_{q+1}, ..., t_M \quad (3)$$

Then, similar to the GME, we first obtain the vector representation $\boldsymbol{l}_i$ of each masked trigger e_i using the attention mechanism and then feed it into a softmax layer to predict the probability of each possible event subtype as follows.

$$\boldsymbol{s}_i = \text{softmax}(\boldsymbol{w}_s^\top \boldsymbol{l}_i + \boldsymbol{b}_s) \quad s_t(i, y) = \boldsymbol{s}_i(y) \quad (4)$$

where $\boldsymbol{w}_s, \boldsymbol{b}_s$ are parameters, the y-th element of $\boldsymbol{s}_i$, i.e., $\boldsymbol{s}_i(y)$, is a score indicating e_i's likelihood of belonging to the event subtype y. Since the event triggers are entirely masked, our model has to mine the contextual clues for reasoning. This prevents the model from simply remembering trigger-to-subtype shortcuts but learning the underlying regularities regarding how events are described in texts. Considering that only the local contexts are used in the event encoding, we take $\boldsymbol{l}_i$ as the sentence-level representation of each event mention e_i. During training, we provide a direct supervision signal to the Local Mention Encoder by reducing the cross-entropy loss $\mathcal{L}_s$ of event subtype classification.

3.3 Event Topic Generator

However, in some situations, it is hard to determine coreferent event mentions relying solely on sentence-level and document-level information (Lu and Ng, 2021c), especially for the main events of a document, which commonly elaborate on new aspects or further information (Choubey and Huang, 2018). Hence, even coreferent main events may have discrepant contexts, especially for those long-distance event pairs. It is difficult to judge event coreference by directly measuring the semantical similarities of the event contents and requires more high-level information, such as event topics.

Similar to the LDA-style models, we believe there is an association between the event element distribution $\boldsymbol{d}_i$ and its topic distribution $\boldsymbol{z}_i$. In particular, for each $\boldsymbol{d}_i$, we infer a latent topic distribution $\boldsymbol{z}_i \in \mathbb{R}^T$, where T denotes the number of topics. Here we take all the verbs and entities in the local context around e_i as event elements and then construct the corresponding BoW (Bag-of-Words) representation $\boldsymbol{d}_i \in |V|$. Although some studies (Choubey and Huang, 2018; Lu and Ng, 2020) have explored the event topics, most directly adopt the traditional LDA models. Inspired by recent neural topic models (NTMs) (Xu et al., 2019; Cao et al., 2021), we propose an event topic generator ETG based on the Variational AutoEncoder (VAE) (Kingma and Welling, 2013).

We believe coreferent event mentions would have similar topic distributions. As such, different from most neural topic models that assume topics follow normal distributions, we make event topics follow the von Mises-Fisher (vMF) spherical distribution. Let $\boldsymbol{\xi} \in \mathbb{R}^T$ be the parameter vector, the vMF distribution is $p(\boldsymbol{x}) = C_{T,\kappa} e^{\kappa\langle\boldsymbol{\mu},\boldsymbol{x}\rangle}$, where $\boldsymbol{\mu} = \boldsymbol{\xi}/\|\boldsymbol{\xi}\|, \kappa = \|\boldsymbol{\xi}\|, C_{T,\kappa} = 1/Z_{T,\|\boldsymbol{\xi}\|}$. Since vMF distribution is measured by cosine similarity of $\boldsymbol{\mu}$ and $\boldsymbol{x}$, the generated event topic distributions naturally fit the intrinsic correlation of coreference. Similar to NTMs, we interpret ETG as a VAE: a neural encoder $p(\boldsymbol{z}|\boldsymbol{d})$ first compresses the event element distribution $\boldsymbol{d}_i$ into a continuous hidden vector $\boldsymbol{z}_i$ as the latent topic, and then an FFNN decoder $g(\boldsymbol{z})$ restores $\boldsymbol{z}_i$ to $\boldsymbol{d}_i$. The inference network $p(\boldsymbol{z}|\boldsymbol{d})$ is defined as follows.

$$\boldsymbol{\mu} = \frac{\boldsymbol{\xi}}{\|\boldsymbol{\xi}\|} \quad \boldsymbol{\xi} = f_\mu\Big(\text{ReLU}\big(f_h(\boldsymbol{d})\big)\Big) \quad (5)$$

where $f_\mu(\cdot), f_h(\cdot)$ are single layer FFNNs. For each event element distribution $\boldsymbol{d}_i$ of the mention e_i, the inference network generates its own $\boldsymbol{\mu}_i$ that parameterize a vMF distribution $p(\boldsymbol{z}|\boldsymbol{d}) = C_{T,\kappa} e^{\kappa\langle\boldsymbol{\mu}(\boldsymbol{d}),\boldsymbol{z}\rangle}$, and we further sample the latent topic $\boldsymbol{z}_i$. To reduce the variance in the stochastic estimation and speed up sampling, we follow (Rezende et al., 2014; Su, 2021) to sample $\boldsymbol{z}$ by the reparametric and pre-sampling methods and sample $\boldsymbol{\epsilon} \sim \mathcal{N}(\boldsymbol{0}, \mathbf{I})$ as follows.

$$\boldsymbol{z} = \boldsymbol{w}\boldsymbol{\mu} + \sqrt{1 - \boldsymbol{w}^2}\boldsymbol{\nu} \quad (6)$$

$$\boldsymbol{w} \sim e^{\kappa \boldsymbol{w}}(1 - \boldsymbol{w}^2)^{\frac{T-3}{2}} \quad \boldsymbol{\nu} = \frac{\boldsymbol{\epsilon} - \langle\boldsymbol{\epsilon}, \boldsymbol{\mu}\rangle\boldsymbol{\mu}}{\|\boldsymbol{\epsilon} - \langle\boldsymbol{\epsilon}, \boldsymbol{\mu}\rangle\boldsymbol{\mu}\|} \quad (7)$$

We hope our ETG can reconstruct the original input $\boldsymbol{d}$ as much as possible using the topic distribution $\boldsymbol{z}$ while adding noise to the result generated by the encoder to enhance the robustness of the decoder. As such, the loss function of ETG is defined as follows.

$$\mathcal{L}_t = \mathbb{E}_{\boldsymbol{z} \sim p(\boldsymbol{z}|\boldsymbol{d})}[-\log q(\boldsymbol{d}|\boldsymbol{z})] + \text{KL}(q(\boldsymbol{z})||p(\boldsymbol{z}|\boldsymbol{d})) \quad (8)$$

where $q(\boldsymbol{z})$ is a uniform spherical distribution, i.e. $\kappa = 0$. It is worth mentioning that reducing the reconstruction loss can make the decoder have the generative ability. We calculate the reconstruction loss by calculating the MSE between the BoW representation $\boldsymbol{d}_i$ and $\widehat{\boldsymbol{d}}_i$ reconstructed by the decoder. Here we take $\kappa \neq 0$ as a hyperparameter such that KL (Kullback-Leibler) divergence $\kappa \left\langle \boldsymbol{\mu}(\boldsymbol{d}), \mathbb{E}_{\boldsymbol{z}\sim p(\boldsymbol{z}|\boldsymbol{d})}[\boldsymbol{z}] \right\rangle + \log C_{T,\kappa} - \log C_{T,0}$ becomes a constant greater than zero. As such, the loss of ETG can be further simplified as follows.

$$\mathcal{L}_t = \|\boldsymbol{d} - g(\boldsymbol{z})\|^2 \tag{9}$$

Since ETG is an unsupervised model, we only need to input the BoW representations $\boldsymbol{d}_i$ during training. Given an event element distribution $\boldsymbol{d}_i$, our ETG can infer its latent distribution $\boldsymbol{z}_i$ as the corresponding topic-level representation.

3.4 Tensor Matching Scorer

After obtaining the document-level, sentence-level, and topic-level embeddings of each event mention e_i, we concatenate them to get the final event representation $\boldsymbol{e}_i = [\boldsymbol{g}_i; \boldsymbol{l}_i; \boldsymbol{z}_i] \in \mathbb{R}^d$. Then, the most direct way is to feed the event pair representation $[\boldsymbol{e}_i; \boldsymbol{e}_j]$ into a softmax layer to predict coreference. Inspired by the tensor networks used in discourse relation recognition (Xu et al., 2019; Liu et al., 2020b), we utilize two tensor matching methods to capture semantic interactions between event mentions: (i) element-wise product, i.e. $\boldsymbol{e}_i \circ \boldsymbol{e}_j$, which is widely used in recent ECR studies (Lai et al., 2021; Lu and Ng, 2021a; Tran et al., 2021); (ii) multi-perspective cosine similarity as follows.

$$\begin{aligned}\mathrm{Cos}(\boldsymbol{e}_i, \boldsymbol{e}_j) = [&\cos(\boldsymbol{W}^c_{1,:} \circ \boldsymbol{e}_i, \boldsymbol{W}^c_{1,:} \circ \boldsymbol{e}_j),\\ &..., \cos(\boldsymbol{W}^c_{s,:} \circ \boldsymbol{e}_i, \boldsymbol{W}^c_{s,:} \circ \boldsymbol{e}_j)]\end{aligned} \tag{10}$$

where $\boldsymbol{W}^c \in \mathbb{R}^{s\times d}$ is the parameter and s is the number of perspectives. To decrease the computational cost, we first reduce the event embedding dimension to v by an FFNN, and then adopt tensor factorization (Pei et al., 2014), using two low-rank matrices $\boldsymbol{P} \in \mathbb{R}^{s\times r}, \boldsymbol{Q} \in \mathbb{R}^{r\times v}$ to approximate $\boldsymbol{W}^c \Rightarrow \boldsymbol{P} \cdot \boldsymbol{Q}$, and $r \ll v$. In this way, we can set more cosine matching perspectives.

Finally, we concatenate the representations of the event pair $\boldsymbol{e}_i, \boldsymbol{e}_j$ and their matching vectors $\boldsymbol{e}_i \circ \boldsymbol{e}_j, \mathrm{Cos}(\boldsymbol{e}_i, \boldsymbol{e}_j)$ and then send them to a softmax layer to predict their coreference. During training, we reduce the cross-entropy loss of coreference judgment $\mathcal{L}_c$ to optimize parameters. After that, we employ a greedy iterative clustering algorithm to create final clusters as follows: we first take all events individually as the initial clusters and then merge any two clusters as long as there is an event pair from the two clusters predicted to be coreferent. Repeat this process until no merging is possible.

3.5 Training

To simultaneously update the parameters in all components, we jointly tackle the subtype recognition, topic modeling, and the coreference classification, and define the overall loss function as follows.

$$\mathcal{L} = \sum_{i\in\{s,t,c\}} \log(1 + \mathcal{L}_i) \tag{11}$$

This way, the optimizer can automatically regulate the balances among these three modules by weights $\frac{1}{1+\mathcal{L}_i}, i \in \{s,t,c\}$ without manually setting the trade-off parameters. To prevent overfitting, a dropout operation is performed on the output of the GME and the LME.

4 Experimentation

In this section, we first introduce the experimental settings and then report the experimental results.

4.1 Experimental Settings

Following previous work (Lu and Ng, 2021c), we train our model on the KBP 2015 and KBP 2016 datasets (Mitamura et al., 2015, 2016) and evaluate the model on the KBP 2017 dataset (Mitamura et al., 2017). In particular, the KBP 2015 and KBP 2016 datasets (i.e., LDC2015E29, E68, E73, E94, and LDC2016E64) include 817 annotated documents, and the KBP 2017 contain 167 documents. Following Lu and Ng (2021c), we use the same 735 documents for training and the remaining 82 for parameter tuning.

The Stanford CoreNLP toolkit[2] is employed to recognize the named entities and verbs in the documents. Since our work focuses on event coreference resolution, we simply build a Longformer-based sequence labeling model to identify triggers, using the BIO schema to label the event subtype of each token. Finally, we report the ECR performance using the official Reference Coreference Scorer[3], which employs four coreference metrics, including MUC (Vilain et al., 1995), B^3 (Bagga and Baldwin, 1998), CEAF$_e$ (Luo, 2005), BLANC (Recasens and Hovy, 2011), and the unweighted average of their F1 scores (AVG-F). Besides, Micro-F1 is used to evaluate the performance of trigger

[2] https://github.com/stanfordnlp/CoreNLP

[3] https://github.com/conll/reference-coreference-scorers

Model	**Event Coreference Resolution**					**Trigger Detection**		
	MUC	B^3	CEAF$_e$	BLANC	AVG-F	P	R	F1
Sentence-level								
Huang2019	35.7	43.2	40.0	32.4	36.8	56.8	46.4	51.1
BERT	36.5	54.4	55.8	37.3	46.0	63.0	58.1	60.4
RoBERTa	36.0	54.8	55.6	37.3	45.9	63.0	58.1	60.4
Segment-level								
Lu&Ng2020	37.1	44.5	40.0	29.9	37.9	64.5	46.9	54.3
Lu&Ng2021	45.2	54.7	53.8	38.2	48.0	71.6	58.7	64.5
Document-level								
Ours	**46.2**	**57.4**	**59.0**	**42.0**	**51.2**	63.0	58.1	60.4

Table 1: Performances of different models on the KBP 2017 dataset.

detection, where a trigger is considered correctly detected if it has an exact match with a gold trigger in terms of boundary and event subtype.

In our experiments, we use a Longformer-large model in GME and a BERT-base model in LME. To alleviate the data sparseness, we also take verbs and entities in the preceding and following sentences as elements of the current event mention and limit the vocabulary to the top 500 most frequent words, i.e., $|V| = 500$. For LME, we truncate the sentence centered on the trigger to make the maximum local context length 256. In ETG, the number of topics and κ are set to 32 and 20, and the number of neurons in the single-layer FFNN $f_\mu(\cdot), f_h(\cdot)$ are set to 32 and 64, respectively. In addition, the generator g is implemented by a two-layer network with a hidden layer size of 64. In TMS, the number of neurons v in the dimension reduction FFNN is 64, the number of matching perspectives s is set to 128, and r of the tensor factorization is set to 4. For training, we use document-sized mini-batches, i.e., each batch includes only one document with all corresponding event mentions, and apply the Adam optimizer with a learning rate of 1e-5 to update all the parameters.

4.2 Experimental Results

We compare our proposed model with the SOTA models in the same evaluation setting, including a classical pairwise model **Huang2019** (Huang et al., 2019) and two joint models: (i) **Lu&Ng2020** (Lu and Ng, 2020), which with a supervised topic model, (ii) **Lu&Ng2021** (Lu and Ng, 2021b), which joint modeling six event tasks. In addition, we build two pairwise baselines: **BERT** and **RoBERTa** model, using BERT/RoBERTa-large as text encoder. Specifically, it first sends the concatenated event sentence (host the event mention) pairs into the encoder and then obtains the two event trigger representations $\boldsymbol{v}_i, \boldsymbol{v}_j$ by an attention mechanism. Finally, the combined feature vector $[\boldsymbol{v}_i; \boldsymbol{v}_j; \boldsymbol{v}_i \circ \boldsymbol{v}_j]$ is used to judge coreference.

Table 1 reports the performances of the five baselines and our model on KBP 2017, and the results show our model outperforms the best Lu&Ng2021 significantly, with the improvements of 3.2 in the average score AVG-F. This result indicates the effectiveness of our proposed model in resolving event coreference.

Compared with Huang2019, BERT and RoBERTa not only update the text encoder but also input the more accurate predicted triggers; hence, the ECR performance has a substantial increase. Benefiting from the event encoding changed from sentence-level to segment-level context, chunk-based methods Lu&Ng2020 and Lu&Ng2021 outperform the three pairwise models in cases with similar trigger detection performance. This proves that incorporating more contextual information in event encoding is important for the ECR task. Lu&Ng 2021 jointly models more related event tasks compared to Lu&Ng2020, thereby greatly improving the performance of trigger detection and ECR. However, due to the constraints of event encoding, these methods only predict coreferences based on local context. In contrast, our model uses document-level event embeddings based on full-text context and additionally obtains local sentence-level and topic-level representations, thus gaining the best performance.

5 Analysis and Discussion

In this section, we first analyze the impact of document-level, sentence-level and topic-level information, and then discuss the Impact of different tensor matching.

5.1 Impact of Document-level Information

To further evaluate the contribution of document-level information in the GME, we construct some variants of our ECR model: (i) Pairwise(BERT) and Pairwise(RoBERTa): replacing GME with the classical pairwise encoder, which separately predicts event coreference for every event mention pair based only on sentence-level context; (ii) Chunk(BERT) and Chunk(RoBERTa)[4]: following (Lu and Ng, 2021b; Tran et al., 2021), each document is split into multiple chunks and then separately encoded with BERT/RoBERTa, obtaining event mention representations based on segment-level context. The results are shown in Table 2.

Model	MUC	B3	CEA.	BLA.	AVG
Pairwise(BERT)	35.3	54.4	55.8	36.6	45.5
Pairwise(RoBERTa)	39.0	54.3	56.4	38.6	47.1
Chunk(BERT)	38.4	54.9	55.4	37.9	46.7
Chunk(RoBERTa)	39.6	55.2	56.9	38.5	47.6
Ours	46.2	57.4	59.0	42.0	51.2

Table 2: Results using different event encodings.

Table 2 indicates that, benefiting from learning representations in a wide range of segment-level contexts, the chunk-based models perform better than those corresponding pairwise models based only on sentence-level context; the Chunk(BERT) and Chunk(RoBERTa) models are improved by 1.2 and 0.5, respectively. However, these models essentially judge coreferences using only local context. In contrast, with the introduction of document-level context, our model using GME can learn event representations based on full-text context, thus achieving the best performance.

To deeply compare these different event encodings and exclude the effect of the clustering step, we also report the event-pair classification F1-score in Table 3: (1) ALL: results of all (predicted) event mention pairs; (2) results of event mention pairs of different distances: (i) SAME: in the same sentence; (ii) ADJ: in adjacent sentences, i.e., sentence distance ≤ 3; (iii) DIST: sentence distance > 3.

The results of the event-pair classification show that the chunk-based models still outperform those corresponding pairwise models. Specifically, incorporating more contextual information can help identify event coreferences among distant event mention pairs, and the performance of the chunk-based BERT model is greatly improved by 2.5 and 4.6 in the ADJ and DIST cases, respectively. This confirms our hypothesis that identifying event coreferences among long-distance event mentions requires more global information. Our ECR model obtains document-level event representations based on full-text context and hence achieves the best performance in all cases, completing an F1 score improvement from 5.5 to 10.7 for all event mention pairs. This verifies the superiority of the global event encoding adopted by our model for ECR.

Model	ALL	SAME	ADJ	DIST
Pairwise(BERT)	29.4	19.8	40.5	25.7
Pairwise(RoBERTa)	34.2	20.1	44.3	31.7
Chunk(BERT)	33.2	22.5	43.0	30.3
Chunk(RoBERTa)	34.6	18.0	44.8	32.1
Ours	40.1	28.4	50.2	37.1

Table 3: Event-pair classification F1-scores of event mention pairs of different distances.

5.2 Impact of Sentence-level and Topic-level Information

To analyze the impact of sentence-level and topic-level information, we take the variants containing only the GME as the Base model and then add our LME and ETG. In addition, we build an event topic model similar to the Simplified Topic Model (Xu et al., 2019), denoted as STM, which assumes that the event topics follow the normal distributions and additionally performs a batch normalization after obtaining the mean representations to avoid KL vanishing. The results are shown in Table 4.

Table 4 shows slight improvements in most ECR metrics after adding sentence-level and topic-level representations. Specifically, the event-pair results indicate that after mining event clues from the local context and providing high-level topic information, the F1-score in the SAME case is greatly improved by 2.6. The reason may be that events located in the same sentence share similar elements and context; thus, injecting local event cues and topic-level information in representations can further attract the model's attention to local detail cues. Though, these approaches interfere with identifying adjacent and distant event pairs, the F1-score declines by 0.6 and 1.7 in ADJ and DIST cases, respectively.

We also try applying our LME to classical pairwise models, which concatenate the event representations obtained by BERT/RoBERTa and our sentence-level event representations, i.e., $e_i =$

[4] Due to the large memory usage of the Chunk-based models, the Pairwise and the Chunk-based models here use the Base version of BERT/RoBERTa as the text encoder.

Model	**MUC**	**B3**	**CEA.**	**BLA.**	**AVG**	**SAME**	**ADJ**	**DIST**
Base	45.4	57.3	58.7	42.2	50.9	25.8	50.8	38.8
+Local	45.8	57.5	59.1	42.1	51.1	27.5	50.7	37.2
+Local&ETG	46.2	57.4	59.0	42.0	51.2	28.4	50.2	37.1
+Local&STM	45.0	57.2	59.2	42.0	50.9	25.5	50.3	36.9

Table 4: Results after adding sentence-level and topic-level information.

$[\boldsymbol{v}_i; \boldsymbol{l}_i]$. Then, using the combined feature vector $[\boldsymbol{e}_i; \boldsymbol{e}_j; \boldsymbol{e}_i \circ \boldsymbol{e}_j]$ to predict coreferences. The ECR results in Table 5 show that although the pairwise models are entirely based on local context, providing local clues can further improve them. This confirms the effectiveness of our proposed trigger-mask mechanism in mining event cues.

Model	**MUC**	**B3**	**CEA.**	**BLA.**	**AVG**
BERT	36.5	54.4	55.8	37.3	46.0
RoBERTa	36.0	54.8	55.6	37.3	45.9
BERT+Local	37.6	55.1	57.1	38.5	47.1
RoBERTa+Local	39.0	55.8	58.0	39.6	48.1

Table 5: ECR results after applying our Local Mention Encoder to pairwise models.

Table 4 shows that applying ETG can improve the MUC by 0.4 with a slight impact on other metrics. Since the MUC metric only rewards successful identification of event links and the topic information is mainly to help identify coreferences of main events, we evaluate the event-pair results of event mentions on different length chains. Here we naively select event chains with lengths greater than ten as main chains. A singleton is considered correctly recognized if it is predicted non-coreferent with all other event mentions. Table 6 reports the results, where Base is a variant of our model, which removes ETG.

Model	**Single**	$2 \leq$ **length** < 10	**length** ≥ 10
Base	48.7	75.0 / 39.6 / 51.9	93.3 / 35.4 / 51.4
Base+ETG	47.8	73.0 / 39.8 / 51.5	93.8 / 36.1 / 52.1
Base+STM	48.0	73.1 / 39.1 / 51.0	94.1 / 35.9 / 52.0

Table 6: Event-pair results (P/R/F1) and singleton identification accuracy after adding topic-level information.

Table 6 shows that adding topic-level representations improves the coreference judgment of event mentions on main chains, the ETG and STM increase the event-pair classification F1-score by 0.7 and 0.6, respectively. However, this somewhat impairs the coreference determination of short-chain events and singletons. Interestingly, the results also show that although the overall performance of STM is low, the normal distribution is still an option for modeling event topics.

5.3 Impact of Different Tensor Matching

Results in Table 1 show that document-level event representation with simple tensor matching can significantly improve ECR performance. To evaluate the impact of tensor matching, we let the variant that directly concatenates the two event representations as Base, and compare it with models using the following three matching methods: (i) *Prod*: element-wise product, i.e., $\boldsymbol{e}_i \circ \boldsymbol{e}_j$; (ii) *Diff*: absolute element-wise difference, i.e., $|\boldsymbol{e}_i - \boldsymbol{e}_j|$, adopt by popular sentence vector models (Conneau et al., 2017; Reimers and Gurevych, 2019); (iii) *Cos*: our proposed factored multi-perspective cosine similarity. The results are shown in Table 7.

Table 7 shows that applying tensor matching can significantly improve ECR performance. In particular, besides applying the widely used element-wise product, Base+Prod+Cos adopted by our model also uses *Cos* to calculate similarities from various perspectives, thereby achieving the best performance and significantly improving AVG-F by 5.8 compared to the Base model.

Table 7 also reports the event-pair classification results (P/R/F1). It shows that, compared with Base+Prod, Base+Prod+Cos can capture event interactions from more perspectives; thereby, the F1-score improved by 1.0, 0.6, and 0.9 in the SAME, ADJ, and DIST cases, respectively. Especially for distant events with varied contexts, which requires more semantic clues to judge coreference. Since coreferent events have more minor differences $\|\boldsymbol{e}_i - \boldsymbol{e}_j\|$ than non-coreferent events, adding *Diff* can reinforce the clustering tendency. This can be seen as adding a constraint like attraction and repulsion loss (Kriman and Ji, 2021), strengthening the cohesion of event representations, thus improving recalls. It is worth mentioning that we also implement a contrastive learning loss to enhance event representations, showing similar performance to adding *Diff*. However, compared with

Model	MUC	B3	CEA.	BLA.	AVG	ALL	SAME	ADJ	DIST
Base	36.7	54.9	55.3	34.7	45.4	27.4 / 27.8 / 27.6	12.7 / 08.6 / 10.3	35.8 / 34.4 / 35.1	25.3 / 27.0 / 26.1
Base+Prod	45.4	57.0	58.6	41.2	50.5	40.4 / 38.3 / 39.3	32.5 / 23.7 / 27.4	51.4 / 47.8 / 49.6	36.7 / 35.8 / 36.2
Base+Prod+Cos	46.2	57.4	59.0	42.0	51.2	41.8 / 38.5 / 40.1	35.3 / 23.7 / 28.4	53.6 / 47.2 / 50.2	37.9 / 36.3 / 37.1
Base+Prod+Diff	45.0	56.7	58.9	41.4	50.5	39.3 / 39.8 / 39.6	30.1 / 25.4 / 27.6	51.6 / 48.2 / 49.9	35.6 / 37.8 / 36.7
Base+Prod+Diff+Cos	44.4	56.5	58.6	41.2	50.2	39.3 / 38.7 / 39.0	28.3 / 22.4 / 25.0	52.0 / 47.4 / 49.6	35.6 / 36.7 / 36.1

Table 7: Results of ECR and event-pair classification using different matching methods.

adding *Cos*, this method would reduce the recognition precision, with the F1 scores decreased by 0.8, 0.3, and 0.4 in the SAME, ADJ, and FAR cases, respectively. Even if the model applies both *Diff* and *Cos*, it cannot improve further.

6 Conclusion

In this paper, we first apply a Longformer-based encoder to obtain the document-level event embeddings based on full-text context and an encoder with a trigger-mask mechanism to learn sentence-level event embeddings based on local context. In addition, we propose an event topic generator to infer the latent topic-level representations. Finally, using the above event embeddings, we employ a multiple tensor matching method to capture their interactions at the document, sentence, and topic levels. Experimental results on KBP 2017 dataset show that our proposed model outperforms previous SOTA methods. In future work, we will continue to study how to mine the connections between event mentions from more aspects.

Limitation

Although our method is simple yet effective, it still suffers from two obvious shortcomings. First, since our model adopts the pipeline framework, we need to separately train models for trigger detection and coreference prediction, which inevitably bring error propagation. We also try constructing a joint version of our model that employs the same Longformer (of Global Mention Encoder) to identify triggers and judge coreferences. However, this method resulted in a significant performance drop. Therefore, how to design the joint modeling of trigger detection and coreference prediction for our ECR model is still an unsolved problem. Second, our full model applies both a Longformer and a Bert model to learn event representations. Even if we use document-sized mini-batches, the training hardware requirements (especially graphics memory) are still high. Hence, how to refine the model structure is another focus of our future work.

Acknowledgements

We thank the anonymous reviewers for their valuable comments and suggestions. This work was supported by the National Natural Science Foundation of China (Nos. 61836007, 62276177, and 62006167) and funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD).

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*, pages 563–566.

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.

Iz Beltagy, Matthew E Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*.

Pengfei Cao, Yubo Chen, Yuqing Yang, Kang Liu, and Jun Zhao. 2021. Uncertain local-to-global networks for document-level event factuality identification. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2636–2645.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 2913–2920.

Zheng Chen and Heng Ji. 2009. Graph-based event coreference resolution. In *Proceedings of the 2009 Workshop on Graph-based Methods for Natural Language Processing, ACL-IJCNLP 2009*, pages 54–57.

Zheng Chen, Heng Ji, and Robert M Haralick. 2009. A pairwise event coreference model, feature impact and evaluation for event coreference resolution. In *Proceedings of the Workshop on Events in Emerging Text Types*, pages 17–22.

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.

Prafulla Kumar Choubey and Ruihong Huang. 2018. Improving event coreference resolution by modeling correlations between event coreference chains and document topic structures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 485–495.

Prafulla Kumar Choubey, Aaron Lee, Ruihong Huang, and Lu Wang. 2020. Discourse as a function of event: Profiling discourse structure in news articles around the main event. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5374–5386.

Prafulla Kumar Choubey, Kaushik Raju, and Ruihong Huang. 2018. Identifying the most dominant event in a news article by mining event coreference relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 340–345.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

Agata Cybulska and Piek Vossen. 2015. "bag of events" approach to event coreference resolution. supervised classification of event templates. *International Journal of Computational Linguistics and Applications*, 6(2):11–27.

Yin Jou Huang, Jing Lu, Sadao Kurohashi, and Vincent Ng. 2019. Improving event coreference resolution by learning argument compatibility from unlabeled data. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 785–795.

Kian Kenyon-Dean, Jackie Chi Kit Cheung, and Doina Precup. 2018. Resolving event coreference with supervised representation learning and clustering-oriented regularization. In *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, pages 1–10.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Sebastian Krause, Feiyu Xu, Hans Uszkoreit, and Dirk Weissenborn. 2016. Event linking with sentential features from convolutional neural networks. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 239–249.

Samuel Kriman and Heng Ji. 2021. Joint detection and coreference resolution of entities and events with document-level context aggregation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: Student Research Workshop*, pages 174–179.

Tuan Lai, Heng Ji, Trung Bui, Quan Hung Tran, Franck Dernoncourt, and Walter Chang. 2021. A context-dependent gated module for incorporating symbolic semantics into event coreference resolution. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3491–3499.

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*, pages 489–500.

I-Ta Lee, Maria Leonor Pacheco, and Dan Goldwasser. 2020. Weakly-supervised modeling of contextualized event embedding for discourse relations. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4962–4972.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 236–246.

Jian Liu, Yubo Chen, Kang Liu, Yantao Jia, and Zhicheng Sheng. 2020a. How does context matter? on the robustness of event detection with context-selective mask generalization. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2523–2532.

Xin Liu, Jiefu Ou, Yangqiu Song, and Xin Jiang. 2020b. On the importance of word and sentence representation learning in implicit discourse relation classification. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, pages 3830–3836.

Zhengzhong Liu, Jun Araki, Eduard H Hovy, and Teruko Mitamura. 2014. Supervised within-document event coreference using information propagation. In *Proceedings of the 9th International Conference on Language Resources and Evaluation*, pages 4539–4544.

Jing Lu and Vincent Ng. 2017. Joint learning for event coreference resolution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 90–101.

Jing Lu and Vincent Ng. 2018. Event coreference resolution: A survey of two decades of research. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 5479–5486.

Jing Lu and Vincent Ng. 2020. Event coreference resolution with non-local information. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 653–663.

Jing Lu and Vincent Ng. 2021a. Span-based event coreference resolution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13489–13497.

Jing Lu and Vincent Ng. 2021b. 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*, pages 4504–4514.

Jing Lu and Vincent Ng. 2021c. Conundrums in event coreference resolution: Making sense of the state of the art. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1368–1380.

Jing Lu, Deepak Venugopal, Vibhav Gogate, and Vincent Ng. 2016. Joint inference for event coreference resolution. In *Proceedings of the 26th International Conference on Computational Linguistics*, pages 3264–3275.

Yaojie Lu, Hongyu Lin, Jialong Tang, Xianpei Han, and Le Sun. 2022. End-to-end neural event coreference resolution. *Artificial Intelligence*, 303:103632.

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.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2015. Overview of TAC KBP 2015 event nugget track. In *Proceedings of the Text Analysis Conference*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2016. Overview of tac-kbp 2016 event nugget track. In *Proceedings of the Text Analysis Conference*.

Teruko Mitamura, Zhengzhong Liu, and Eduard H Hovy. 2017. Overview of tac kbp 2015 event nugget track. In *Events Detection, Coreference and Sequencing: What's Next? Overview of the TAC KBP 2017 Event Track*.

Thien Huu Nguyen, Adam Meyers, and Ralph Grishman. 2016. New york university 2016 system for kbp event nugget: A deep learning approach. In *Proceedings of the Text Analysis Conference*.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 293–303.

Haoruo Peng, Yangqiu Song, and Dan Roth. 2016. Event detection and co-reference with minimal supervision. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 392–402.

Duy Phung, Tuan Ngo Nguyen, and Thien Huu Nguyen. 2021. Hierarchical graph convolutional networks for jointly resolving cross-document coreference of entity and event mentions. In *Proceedings of the Fifteenth Workshop on Graph-Based Methods for Natural Language Processing*, pages 32–41.

Marta Recasens and Eduard Hovy. 2011. Blanc: Implementing the rand index for coreference evaluation. *Natural language engineering*, 17(4):485–510.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. 2014. Stochastic backpropagation and approximate inference in deep generative models. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1278–1286.

Jianlin Su. 2021. Variational autoencoders (7): Vae on a sphere (vmf-vae).

Meihan Tong, Bin Xu, Shuai Wang, Yixin Cao, Lei Hou, Juanzi Li, and Jun Xie. 2020. Improving event detection via open-domain trigger knowledge. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 5887–5897.

Hieu Minh Tran, Duy Phung, and Thien Huu Nguyen. 2021. 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*, pages 4840–4850.

Marc Vilain, John D Burger, John Aberdeen, Dennis Connolly, and Lynette Hirschman. 1995. A model-theoretic coreference scoring scheme. In *Proceedings of the 6th Message Understanding Conference*.

Sheng Xu, Peifeng Li, Fang Kong, Qiaoming Zhu, and Guodong Zhou. 2019. Topic tensor network for implicit discourse relation recognition in chinese. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 608–618.

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.

Yutao Zeng, Xiaolong Jin, Saiping Guan, Jiafeng Guo, and Xueqi Cheng. 2020. Event coreference resolution with their paraphrases and argument-aware embeddings. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3084–3094.