Joint Document-Level Event Extraction via Token-Token Bidirectional Event Completed Graph

Qizhi Wan[1,2], **Changxuan Wan**[1,2*], **Keli Xiao**[3*], **Dexi Liu**[1,2], **Chenliang Li**[4]
Bolong Zheng[5], **Xiping Liu**[1,2], **Rong Hu**[1,2]
[1]Jiangxi Key Lab of Data and Knowledge Engineering
[2]School of Information Management, Jiangxi University of Finance and Economics
[3]College of Business, Stony Brook University
[4]School of Cyber Science and Engineering, Wuhan University
[5]School of Comp. Scie. & Tech., Huazhong University of Science and Technology
wanqizhi@163.com, wanchangxuan@263.net, keli.xiao@stonybrook.edu

Abstract

We solve the challenging document-level event extraction problem by proposing a joint exaction methodology that can avoid inefficiency and error propagation issues in classic pipeline methods. Essentially, we address the three crucial limitations in existing studies. First, the autoregressive strategy of path expansion heavily relies on the orders of argument roles. Second, the number of events in documents must be specified in advance. Last, unexpected errors usually exist when decoding events based on the entity-entity adjacency matrix. This paper designs a Token-Token Bidirectional Event Completed Graph (TT-BECG) in which the relation *eType-Role*$_1$*-Role*$_2$ serves as the edge type, precisely revealing which tokens play argument roles in an event of a specific event type. Exploiting the token-token adjacency matrix of the TT-BECG, we develop an edge-enhanced joint document-level event extraction model. Guided by the target token-token adjacency matrix, the predicted token-token adjacency matrix can be obtained during model training. Then, the event records in a document are decoded based on the predicted matrix, including the graph structure and edge-type decoding. Extensive experiments are conducted on two public datasets, and the results confirm the effectiveness of our method and its superiority over the state-of-the-art baselines.

1 Introduction

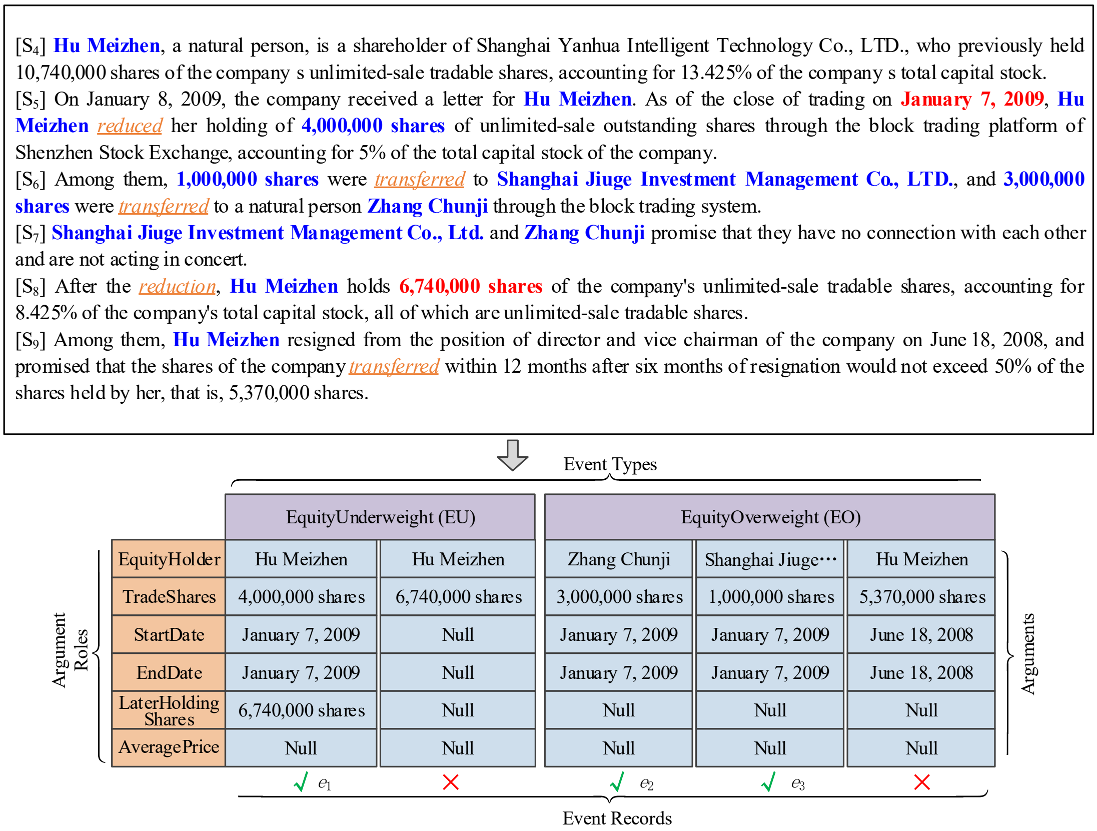
Document-level event extraction aims to recognize events in a document with pre-defined types and corresponding event arguments, which includes entity extraction, entity-event correlation mapping, event type recognition, and argument role judgment. Sentence-level event extraction approaches (Sha et al., 2018; Yang et al., 2019; Lu et al., 2021; Wan et al., 2021, 2023a) are difficult to deal with the problem of arguments across sentences. Also, a document usually contains multiple events without clear boundaries, and the corresponding descriptions may interact together, increasing the challenges of extracting event information accurately.

Figure 1 demonstrates a document example where we can summarize the following challenges in event extraction. (1) Arguments across sentences. The event e_1 of EU (EquityUnderweight) type is triggered by the “reduced” with most of the arguments in S$_5$, yet the argument acting as “LaterHoldingShares” is scattered in S$_8$, and the information describing other events are in S$_6$ and S$_7$; that is, the descriptions of different events are mixed together. (2) Multiple events. We must determine the accurate number of events. Also, considering that tokens reflecting the meaning of “reduce” appear more than once, it will be a problem to determine that there is only one event of the EU type. (3) More noise. Not all entities with the same characteristics act as arguments. For example, both “January 8, 2009” and “January 7, 2009” in S$_5$ are time entities. The former does not act as the argument, while the latter does. Along this line, two strategies have been adopted in existing document-level event extraction models.

Previous studies on document-level event extraction generally adopted the pipeline pattern (Zheng et al., 2019; Xu et al., 2021; Yang et al., 2021; Huang and Jia, 2021; Zhu et al., 2022; Liang et al., 2022), which decomposes the task into the following sub-tasks: (1) entity extraction (obtaining candidate arguments from a document), (2) event type recognition (judging the event types involving in the document and clarifying the ontology of each event type), and (3) multi-event and corresponding argument identification according to the recognized event types in the sub-task (2). Therefore, error

[*]Corresponding Author.

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10481–10492
July 9-14, 2023 ©2023 Association for Computational Linguistics



[S4] **Hu Meizhen**, a natural person, is a shareholder of Shanghai Yanhua Intelligent Technology Co., LTD., who previously held 10,740,000 shares of the company s unlimited-sale tradable shares, accounting for 13.425% of the company s total capital stock.
[S5] On January 8, 2009, the company received a letter for **Hu Meizhen**. As of the close of trading on **January 7, 2009**, **Hu Meizhen** *reduced* her holding of **4,000,000 shares** of unlimited-sale outstanding shares through the block trading platform of Shenzhen Stock Exchange, accounting for 5% of the total capital stock of the company.
[S6] Among them, **1,000,000 shares** were *transferred* to **Shanghai Jiuge Investment Management Co., LTD.**, and **3,000,000 shares** were *transferred* to a natural person **Zhang Chunji** through the block trading system.
[S7] **Shanghai Jiuge Investment Management Co., Ltd.** and **Zhang Chunji** promise that they have no connection with each other and are not acting in concert.
[S8] After the *reduction*, **Hu Meizhen** holds **6,740,000 shares** of the company's unlimited-sale tradable shares, accounting for 8.425% of the company's total capital stock, all of which are unlimited-sale tradable shares.
[S9] Among them, **Hu Meizhen** resigned from the position of director and vice chairman of the company on June 18, 2008, and promised that the shares of the company *transferred* within 12 months after six months of resignation would not exceed 50% of the shares held by her, that is, 5,370,000 shares.

	EquityUnderweight (EU)		EquityOverweight (EO)		
EquityHolder	Hu Meizhen	Hu Meizhen	Zhang Chunji	Shanghai Jiuge···	Hu Meizhen
TradeShares	4,000,000 shares	6,740,000 shares	3,000,000 shares	1,000,000 shares	5,370,000 shares
StartDate	January 7, 2009	Null	January 7, 2009	January 7, 2009	June 18, 2008
EndDate	January 7, 2009	Null	January 7, 2009	January 7, 2009	June 18, 2008
LaterHolding Shares	6,740,000 shares	Null	Null	Null	Null
AveragePrice	Null	Null	Null	Null	Null
	✓ e_1	✗	✓ e_2	✓ e_3	✗

Figure 1: A document example taken from ChFinAnn (Zheng et al., 2019) that translated from Chinese. The upper part is the original document, and the bottom part shows the gold events (i.e., ticked records, denoted as e_1, e_2, and e_3). The bold tokens (red and blue) represent gold event arguments, and the red tokens are cross-sentence arguments.

propagation exists in these methods.

In terms of implementation strategies, the methods based on graph decoding are mainly divided into entity-based directed acyclic graph (Zheng et al., 2019; Xu et al., 2021; Liang et al., 2022) and pseudo-trigger-aware pruned complete graph (Zhu et al., 2022).

The former employed a path-expanding autoregression strategy but relied heavily on the specified argument role order of a triggered event type, resulting in the pipeline pattern only being adopted and huge training costs. To narrow down candidate entities for argument recognition, the latter established the mappings between entities and events with the idea of the clique. Since triggers are not marked in the corpus, the concept of pseudo-triggers was proposed, and a pruned complete graph was constructed based on the selected pseudo-triggers. Nevertheless, the constructed graph cannot be fully decoded into corresponding event records due to sharing the same pseudo triggers; that is, the gold training target of the model has errors, hence will seriously affect model learning and the final event record decoding based on the predicted graph.

To realize the joint document-level event extraction, this paper devises a *Token-Token Bidirectional Event Completed Graph* (TT-BECG) with the relation *eType-Role*$_1$*-Role*$_2$ as the edge type, accurately revealing which tokens play argument roles in an event of a specific event type. Thus, all tasks for document-level event extraction by the pipeline pattern are integrated into the graph.

Employing the adjacency matrix of TT-BECG, we develop an edge-enhanced joint document-level event extraction model (EDEE). Specifically, based on the adjacency matrix of target TT-BECG (generated according to the corpus), the model is trained to approximate it and obtain the predicted token-token adjacency matrix. All event records can be decoded by the predicted adjacency matrix, including graph structure and edge-type decoding. Therefore, the whole document-level event extraction is transformed into the prediction and decoding task of the token-token adjacency matrix, achieving the joint extraction.

To sum up, the main contributions of this work are threefold.

- We design a token-token bidirectional event completed graph with the relation *eType-Role*$_1$-*Role*$_2$ as the edge type. It can accurately decode which tokens play the argument roles of an event in a specific event type and solve the problems of multi-event and argument cross-sentence, as well as the limitations of previous studies.
- We develop an edge-enhanced joint document-level event extraction framework, which integrates all event extraction tasks involving the pipeline pattern to prevent error propagation. This paper is the first joint event extraction work for the document level.
- Extensive experiments are conducted on two public datasets, and the results confirm the effectiveness of our scheme with a superiority of 15.3∼38.4 and 26.9∼47.5 percentage points, respectively.

2 Methodology

In this paper, the document-level event extraction is converted to a prediction and decoding task for graph structure and edge type. It mainly includes three stages: (1) constructing the target token-token bidirectional event completed graph (TT-BECG) according to the training corpus, along with the corresponding adjacency matrix; (2) designing the model for training and obtaining the predicted token-token adjacency matrix; (3) decoding the predicted adjacency matrix, generating all events and event records contained in documents.

2.1 TT-BECG and Adjacency Matrix

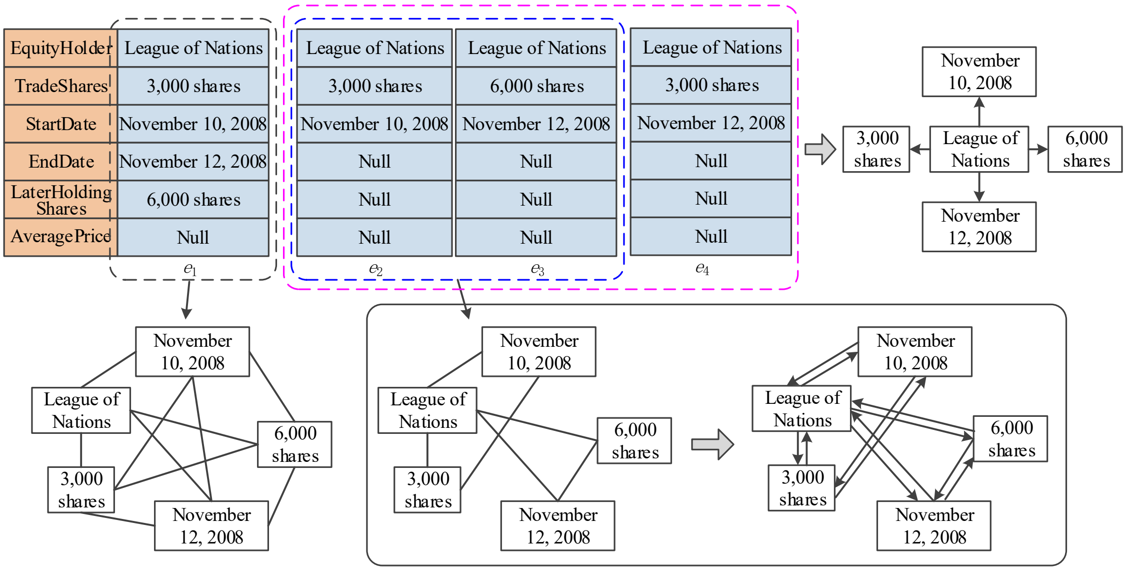
Origin of TT-BECG. Due to sharing the same pseudo triggers, there are errors in the event decoding of Zhu et al. (2022), as shown in Figure 2. When the token "League of Nations" is selected as a pseudo trigger, the entity-entity graphs of the record $\{e_1\}$, $\{e_2, e_3\}$ and $\{e_2, e_3, e_4\}$ are identical (see the upper right part). Meanwhile, decoding events is ambiguous to determine which dotted line box of the event record or any other combinations corresponding to the graph structure. The main reason is the strategy needs to select pseudo triggers and take them as the center. Once the pseudo triggers are identical or partially overlapping, errors will occur when decoding. However, as the number of pseudo triggers increases, the entity-entity graph becomes more complex, and the effect of approximating the target entity-entity matrix decreases rapidly, resulting in invalid argument recognition (Zhu et al., 2022).

To solve these problems, this paper designs a new graph structure. First, we discard the strategy centered on pseudo triggers and correlate all arguments in an event (i.e., construct a completed graph as shown in the bottom left of Figure 2), so that each completed subgraph in the entity-entity graph can accurately decode the entity-event correspondence, solving the multi-event problem. Then, since the undirected entity-entity graph only reveals the association between entities, the edge types of $ent_i \rightarrow ent_j$ and $ent_j \rightarrow ent_i$ are *eType-role*$_i$-*role*$_j$ and *eType-role*$_j$-*role*$_i$, respectively. (ent_i and ent_j represent entities, *eType* is event type, $role_i$ and $role_j$ are argument roles.) That is, the Id tag entered in the corresponding position of the entity-entity adjacency matrix is not the same. Therefore, the edge types in the entity-entity graph should be bidirectional, as shown in the bottom right of Figure 2. Note that, if the same entity acts as an argument for different roles in the same or different events, we treat it as a new entity.

Finally, to realize the joint document-level event extraction, we develop a token-token bidirectional event completed graph with *eType-Role*$_1$-*Role*$_2$ as the edge type. By decoding all the edge types between tokens in each completed subgraph (a completed subgraph corresponds to an event) contained in the graph, it is clear which tokens play the specific argument roles in an event of a specific event type.

Token-Token Adjacency Matrix Construction. To guide the model training, the target token-token adjacency matrix (denoted as TT) needs to be constructed first. Each value in TT is an Id identifier corresponding to the relation *eType-Role*$_1$-*Role*$_2$, where *Role*$_1$ and *Role*$_2$ represent the argument role played by the first token (correspond row in TT) and the second token (correspond column in TT) in the token-token pair of the event type *eType*. The specific construction steps of TT are summarized as follows.

Step 1. Given each argument role (denoted as *role*) of each event type (*eType*), split it into "B_*role*" and "I_*role*" tags, where they are assigned to the head and other position of an argument, respectively, addressing the problem that multiple



	e_1	e_2	e_3	e_4
EquityHolder	League of Nations	League of Nations	League of Nations	League of Nations
TradeShares	3,000 shares	3,000 shares	6,000 shares	3,000 shares
StartDate	November 10, 2008	November 10, 2008	November 12, 2008	November 12, 2008
EndDate	November 12, 2008	Null	Null	Null
LaterHolding Shares	6,000 shares	Null	Null	Null
AveragePrice	Null	Null	Null	Null

Figure 2: Example of clique-based event decoding and the construction process of the token-token bidirectional event completed graph. The upper left shows event records, and the upper right is the corresponding entity-entity graph. The entity "League of Nations" acting as "EquityHolder" role is selected as the pseudo trigger, and pseudo triggers point to non-pseudo triggers within the same event. When the document only contains the event records in the blue dotted line box, the final bidirectional event completed graph is demonstrated at the bottom right.

tokens are combined to act as an argument. Then, all the split role tags of the event type are combined in pairs, along with *eType*, forming the relation *eType-X_role*$_1$*-X_role*$_2$. *X* represents B or I. Finally, all relations of event types are formed into a set (denoted as ***Edges***), and each element in the set is given a sequence number starting from 1 as its Id identifier. In addition, non-argumental tokens are assigned the role tag "O" and added to the ***Edges*** with an Id identifier of 0.

Step 2. For each document in the corpus, any tokens w_i (row *i* in TT) and w_j (column *j* in TT) in arguments of event records are combined, and the corresponding relation is denoted as *eType-X_Role*$_i$*-X_Role*$_j$. The Id identifier of the relation is filled in **TT**[*i*, *j*].

Step 3. If a token plays different roles in the same/different events, it is regarded as a new token. The row and column of TT are each increased by 1, and the new role is filled in the newly added row and column according to the method in step 2.

2.2 EDEE

In the following, we describe our edge-enhanced document-level event extraction model (EDEE). As demonstrated in Figure 3, the framework includes three components: (1) the Embedding Layer, for initializing the semantic embeddings and capturing sequential semantics between tokens by Bi-LSTM network; (2) the Classification Layer, for predicting the label of each token-token pair and generating predicted TT; (3) the TT Decoding, decoding extracted events and event records according to the predicted TT, including graph structure decoding (determine the number of events) and edge type decoding (clarify argument roles of tokens playing in events of a specific event type).

Embedding Layer. In this paper, the BERT (Devlin et al., 2019) is used to initialize token embedding, and the vector of *i*-th token w_i is denoted as $\mathbf{v}_i$. Following previous studies (Xu et al., 2021; Zhu et al., 2022), the entity type is also exploited, and the vector is generated by looking up the randomly initialized embedding table. Then, we concatenate $\mathbf{v}_i$ with the corresponding entity vector and pour them into a Bi-LSTM network to capture the sequential information. The output embedding is denoted as $\mathbf{h}_i$. Finally, any $\mathbf{h}_i$ and $\mathbf{h}_j$ are concatenated to represent the embedding of the *k*-th token-token pair w_i-w_j, forming $\mathbf{h}'_k \in \mathbb{R}^{2d}$, *d* refers to the dimension of $\mathbf{h}_i$.

Classification Layer. The softmax function is adopted to compute the distribution $p(y_k|\theta)$ of the embedding $\mathbf{h}'_k$ over the relation tags:

$$p\left(y_k|\theta\right) = \mathrm{softmax}\left(\mathbf{W}_p\mathbf{h}'_k + b_p\right), \qquad (1)$$

where y_k is the tag of *k*-th token-token pair under the parameter θ, $\mathbf{W}_p$ denotes a weight matrix, and

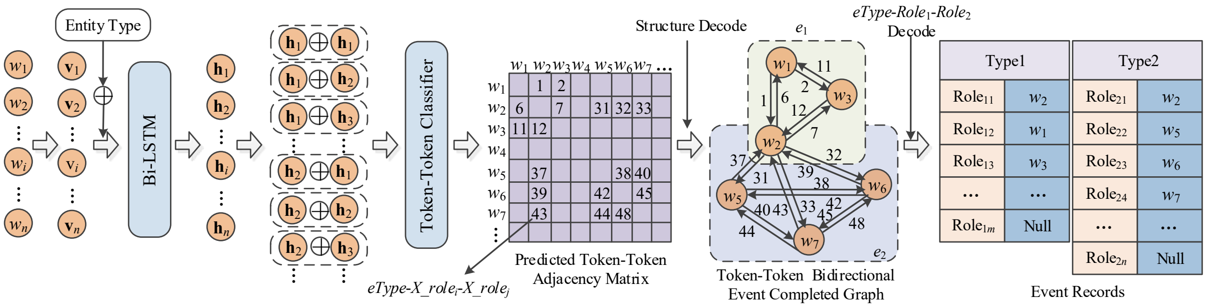


Figure 3: The overall framework. ⊕ represents the concatenation operation. Each Id identifier in the predicted token-token adjacency matrix corresponds to a relation *eType-X_role$_i$-X_role$_j$*, demonstrating the token-token pair w_i-w_j acts as the role X_role_i and X_role_j in the event type *eType*. The unfilled position is 0, indicating that the tokens are not related to each other.

b_p is a bias term. Finally, the tag with the largest probability is chosen as the classification result.

Following previous studies (Chen et al., 2018; Wan et al., 2023a), given that the number of "O" tags is much larger than that of other relation tags, the standard cross-entropy loss with weight is used as our objective function to strengthen the influence of relation tags:

$$J(\theta) = -\sum_{k=1}^{n\times n} \omega_k \log p\left(y_k|\theta\right), \qquad (2)$$

where n is the number of tokens in a document, ω_k is the weight of y_k tag, which can be obtained by the method in Wan et al. (2023a).

2.3 Token-Token Matrix Decode

Through the classifier, the predicted token-token adjacency matrix $\mathrm{TT}^{(\mathrm{p})}$ can be obtained. Then, the graph structure decoding is first implemented based on $\mathrm{TT}^{(\mathrm{p})}$; that is, the edges are established for the tokens in which their Id identifiers of token-token pairs are non-zero in $\mathrm{TT}^{(\mathrm{p})}$, forming the token-token bidirectional event completed graph in Figure 3. A completed subgraph corresponds to an event; thus, the completed graph in Figure 3 can be decoded into two events. Subsequently, the Id identifier of edge type in each subgraph is transformed into *eType-Role*$_1$*-Role*$_2$, and the final event records are parsed according to different relations.

3 Experiments and Results

3.1 Data and Evaluation Metrics

This paper conducted experiments on two public datasets ChFinAnn (Zheng et al., 2019) and DuEE-Fin (Han et al., 2022). ChFinAnn consists of 32,040 documents covering five event types with 35 different kinds of argument roles in total. DuEE-Fin is published by Baidu, including 13 event types and 11,700 documents, and each event record is labeled with the argument and trigger information, while the trigger information is not used in our experiments. We followed the standard split of the two datasets for training/development/test set. The LTP (Che et al., 2021) syntactic tool was used for word segmentation.

Regarding the evaluation metrics, the Precision (***P***), Recall (***R***) and $F1$-score (***F1***) were selected to evaluate the models. Since a document contains enormous tokens and few of them serve as gold argument roles, the model performed well for those who are not the argument (i.e., the tokens marked as "O"). However, calculating the overall $F1$ score can not accurately reflect the recognition effect of arguments. Therefore, the prediction results of the "O" tag were filtered out in the evaluation.

3.2 Hyper-Parameter Setting and Baselines

We chose the Adam optimizer in experiments; set batch size = 1, learning late = 1e-3, dropout = 0.2, and iteration = 15. The embedding dimensions of the token and entity type were set to 768 and 50. The hidden size and layers of Bi-LSTM were set to 200 and 4, respectively. The experimental environment of this paper is Python3.7, PyTorch1.12.0, and NVIDIA GeForce RTX 3090 24G.

To comprehensively evaluate our proposed model (EDEE), we followed previous studies (Yang et al., 2021; Zhu et al., 2022) and compared it with a range of baselines, including state-of-the-art models. **DCFEE** (Yang et al., 2018) developed a key-event sentence detection to extract arguments from the key-event mention and surrounding sentences. The model has two variants: DCFEE-O

Model	Mem.	Time	EF	ER	EU	EO	EP	Avg
DCFEE-O	21.3	192	51.1	83.1	45.3	46.6	63.9	58.0
DCFEE-M	23.5	192	45.6	80.8	44.2	44.9	62.9	55.7
Greedy-Dec	22.2	604.8	58.9	78.9	51.2	51.3	62.1	60.5
Doc2EDAG	23.8	604.8	70.2	87.3	71.8	75.0	77.3	76.3
GIT	23.8	633.6	73.4	90.8	74.3	76.3	77.7	78.5
DE-PPN	23.8	50.0	73.5	87.4	74.4	75.8	78.4	77.9
SCDEE	22.8	39.2	80.4	90.5	75.1	70.1	78.1	78.8
PTPCG	**7.1**	**24**	71.4	**91.6**	71.5	72.2	76.4	76.6
EDEE	12.5	49.8	**97.4**	90.3	**93.2**	**93.4**	**96.2**	**94.1**

Table 1: Main results ($F1$) on ChFinAnn dataset. Mem. refers to the GPU memory, and the units of Memory and Time are G and hours. The time results of the first six baselines are taken from Zhu et al. (2022), and the memory and the other time results are reproduced.

Model	EP	EUP	ER	EU	EO	EC	EB	EM	EF	CL	OB	ED	BC	Avg
DCFEE-O	59.9	61.3	75.1	49.8	36.7	34.3	12.4	37.5	54.1	30.1	50.4	65.7	23.7	45.5
DCFEE-M	48.8	49.8	58.8	36.6	29.5	27.8	17.9	32.0	34.7	24.1	41.4	59.6	25.0	37.4
Greedy-Dec	48.0	65.2	71.4	47.3	38.5	33.8	29.9	42.6	52.5	36.7	52.5	66.5	21.4	46.6
Doc2EDAG	71.8	72.3	79.4	55.4	51.2	34.1	35.2	46.1	57.7	41.1	61.8	72.8	23.7	54.1
GIT	**73.7**	**73.1**	77.7	60.8	48.4	42.7	39.3	49.0	62.0	37.1	59.9	73.8	25.6	55.6
DE-PPN	46.0	52.4	52.1	37.4	29.4	29.1	26.9	33.8	32.2	30.8	28.2	62.7	22.9	37.2
PTPCG	69.0	62.2	87.7	58.3	46.0	47.5	39.0	51.3	66.1	39.8	62.3	76.0	46.4	57.8
EDEE	71.8	69.1	**94.3**	**90.4**	**77.6**	**89.3**	**88.0**	**85.9**	**89.6**	**87.6**	**91.8**	**92.5**	**72.8**	**84.7**

Table 2: Main results ($F1$) on DuEE-Fin dataset. The event types are: EP(Equity Pledge), EUP(Equity UnPledge), ER(Equity Repurchase), EU(Equity Underweight), EO(Equity Overweight), EC(Executive Change), EB(Enterprise Bankrupt), EM(Enterprise Merge), EF(Enterprise Financing), CL(Company Listing), OB(Out Bid), ED(Enterprise Deficit), and BC(Be Called). We reproduce the results of baseline using the open-source codes of Zhu et al. (2022).

produces one event record, and DCFEE-M extracts multiple events. **Doc2EDAG** (Zheng et al., 2019) transformed document-level event extraction as directly filling event tables with entity-based path expanding. **Greedy-Dec** is a simple baseline of Doc2EDAG. **GIT** (Xu et al., 2021) designed a heterogeneous graph-based interaction model to capture global interactions. **DE-PPN** (Yang et al., 2021) proposed a document-level encoder and a multi-granularity decoder to extract all events in parallel. **SCDEE** (Huang and Jia, 2021) introduced the sentence community and assigned all sentences of an event to a sentence community. **PTPCG** (Zhu et al., 2022) constructed the maximal clique by calculating pseudo-triggers and incorporated other common entities to complete the clique.

3.3 Overall Performance

Tables 1 and 2 report our experimental results on the two datasets, where Avg is the average of $F1$ score.

As shown in Tables 1 and Table 2, our EDEE consistently outperforms other baselines on ChFinAnn and Ducc-fin, with its Avg of 94.1% and 84.7%, respectively. The corresponding increases are 15.3∼38.4 and 26.9∼47.5 percentage points. Note that all baselines are pipelined and suffer from serial predictions (e.g., entity extraction), resulting in error propagation. In the following, we provide further analyses to investigate the performance impacts of (1) error propagation for entity extraction, (2) entity-event correspondence error propagation, and (3) intermediate phases of baselines.

Error propagation for entity extraction. By analyzing the results of baselines in each phase, it can be found that there are many errors in entity extraction (Zhu et al., 2022), especially for financial data that contains abundant numerical words (e.g., *money*, *percentage ratio*, and *shares*). A common model for entity extraction is challenging in identifying such entities. Xu et al. (2021) showed that there were more than 10 percentage points of errors in the first five baselines in Table 1, hence would directly affect the subsequent identification performance of the argument role. When the training samples are small, it is insufficient to support the learning of the model in all phases, resulting in unfavorable results.

Model	**EF**		**ER**		**EU**		**EO**		**EP**		**Avg**	
	S.	**M.**	**S.**	**M.**	**S.**	**M.**	**S.**	**M.**	**S.**	**M.**	**S.**	**M.**
DCFEE-O	55.7	38.1	83.0	55.5	52.3	41.4	49.2	43.6	62.4	52.2	60.5	46.2
DCFEE-M	45.3	40.5	76.1	50.6	48.3	43.1	45.7	43.3	58.1	51.2	54.7	45.7
Greedy-Dec	74.0	40.7	82.2	50.0	61.5	35.6	63.4	29.4	78.6	36.5	71.9	38.4
Doc2EDAG	79.7	63.3	90.4	70.7	74.7	63.3	76.1	70.2	84.3	69.3	81.0	67.4
GIT	81.9	65.9	93.0	71.7	82.0	64.1	80.9	70.6	85.0	73.5	84.6	69.2
DE-PPN	82.1	63.5	89.1	70.5	79.7	66.7	80.6	69.6	88.0	73.2	83.9	68.7
PTPCG	83.6	59.9	**93.7**	73.8	77.3	63.6	79.7	62.8	86.1	70.5	84.1	66.1
EDEE	**97.9**	**92.2**	90.4	**87.4**	**97.2**	**93.1**	**93.7**	**85.6**	**98.0**	**91.6**	**95.4**	**90.0**

Table 3: $F1$ score of Single-Event and Multi-Event on ChFinAnn dataset. The value of the first five lines is taken from Xu et al. (2021), and others are taken from the original paper.

Entity-event correspondence error propagation. For SCDEE and PTPCG, in addition to the entity extraction and event type recognition errors, there are also errors in the assignment of sentences to communities and event decoding by clique, respectively. According to Zhu et al. (2022), 14.6 percentage points of errors have been found in the target entity-entity adjacency matrix of PTPCG when decoding events. Thus, there may be more errors based on the predicted matrix output by the model. In the experiment, the precision, recall, and $F1$ score of argument combination are only 40.9%. This indicates that the entities that serve as the arguments of an event are not in the same clique, and each clique includes numerous entities of other cliques (events). These factors are also the primary reason for its Avg indicator being inferior to GIT and SCDEE.

Intermediate phases of baselines. Due to the similar intermediate phases, baselines' performances on Avg are comparative. GIT captured and encoded the structure information in the document-level heterogeneous interaction graph; thus, it outperformed the Doc2EDAG with 2.2 percentage points on Avg. SCDEE benefits from the divided sentence community. Compared with other baselines that treat all entities in a document as candidate arguments, SCDEE narrowed the range of candidate arguments, reducing the difficulty of training the model to determine whether an entity acts as a specified role argument in a given event. Hence, it achieves the best effect in baselines. However, all baselines are pipelined patterns, and the propagation of errors restricts their performances by only about 77%, which is still much lower than the joint model in this paper.

Regarding spatio-temporal efficiency, our model also achieves good results. The model implementation only consumes 49.8 hours and 11.7G GPU memory. Compared with the first five baselines in Table 1, the time cost is significantly reduced. Meanwhile, although the token-token bidirectional event completed graph is oriented to all tokens in the document, the model has fewer intermediate phases and sample network structure, ensuring it the second place in spatio-temporal cost.

To sum up, the excellent effect of EDEE mainly lies in the following factors. (1) A data structure (*eType-Role*$_1$-*Role*$_2$) is designed, which can clarify which tokens play roles in an event of a specific event type, integrating the event type and argument role identification together and ensure the joint event extraction framework implementation. (2) Multi-event decoding strategy based on the token-token bidirectional event completed graph is formulated, accurately decoding all events and event records contained in the graph. (3) A joint extraction framework is developed to prevent the error propagation of pipelined patterns by converting the document-level event extraction into the prediction and decoding task of the token-token adjacency matrix.

4 Additional Analysis and Discussions

To further investigate the impact of each component on event extraction performance, we conducted additional experiments, including single & multiple events and ablation.

4.1 Single & Multiple Event

This section aims to analyze the extraction effects of models when a document contains only one or more events. Table 3 reports the $F1$ score of each model under single-event (S.) and multi-event (M.).

In Table 3, for single-event and multi-event, our model (EDEE) obviously outperforms all baselines in most event types and Avg, verifying that EDEE is as effective in dealing with the single-event and

Ablation	EF	ER	EU	EO	EP	Avg
Ours	97.4	90.3	93.2	93.4	96.2	94.1
w/o Entity Type	58.6	84.0	66.8	62.8	73.5	69.1
w/o Bi-LSTM	45.9	87.5	42.8	24.0	87.4	57.5

Table 4: $F1$ score of ablation.

multi-event separately. Concretely, in the Avg indicator, single-event and multi-event are superior to baselines with 10.8∼40.7 and 20.8∼51.6 percentage points, respectively. Compared with Table 1, it can be seen that the overall effect trend is consistent across baselines. GIT and DE-PPN perform well with a slight distinction, and PTPCG is slightly lower than them.

4.2 Ablation

In addition to the completed graph and the triple relation, the entity type and Bi-LSTM network were exploited in this paper. To verify their validity, we conducted ablation experiments.

As Table 4 shows, meaningful cues and appropriate neural networks are necessary to supplement and update the semantic information of tokens. Both sentence-level (Chen et al., 2015; Nguyen et al., 2016) and document-level (Zheng et al., 2019; Zhu et al., 2022) event extraction encoded the entity type feature, verifying its significance. In this paper, EDEE improved by 25.0 percentage points by adding this information. Consistent with previous work, the Bi-LSTM network can capture the sequence structure information between tokens well, which is conducive to event extraction (Chen et al., 2018; Wan et al., 2023a). Removing Bi-LSTM (line 3) indicates that the embeddings of tokens are not updated enough to capture the semantics between tokens, resulting in a 36.6 percent decrease in Avg.

In summary, the token-token bidirectional event completed graph provides a joint execution strategy for document-level event extraction, and appropriate cues and networks can help capture more semantic information, which is also an indispensable part of the entire framework. However, thanks to the completed graph designed in this paper, the EDEE model only needs a few cues and a simple network structure to achieve excellent results.

5 Related Work

With the corpus release for document-level event extraction, such as ChFinAnn (Zheng et al., 2019) and RAMS (Ebner et al., 2020), this task has attracted more and more attention recently (Lin et al., 2022; Ma et al., 2022; Wan et al., 2023b, 2022). Ebner et al. (2020) designed a span-based argument linking model. A two-step method was proposed for argument linking by detecting cross-sentence arguments (Zhang et al., 2020). Du and Cardie (2020) tried to encode sentence information in a multi-granularity way, and Li et al. (2021) developed a neural event network model by conditional generation. Ma et al. (2022) and Lin et al. (2022) exploited prompts and language models for document-level event argument extraction. Nevertheless, these studies only considered the sub-task of document-level event extraction (i.e., role filler extraction or argument extraction) and ignored the challenge of multi-events (Yang et al., 2021).

Therefore, some other studies focused on the multi-event corpus (ChFinAnn). Yang et al. (2018) extracted events from a key-event sentence and found other arguments from neighboring sentences. Zheng et al. (2019) implemented event extraction following a pre-defined order of argument roles with an entity-based path expansion. Subsequently, Xu et al. (2021) built a heterogeneous interaction graph network to capture global interactions among different sentences and entity mentions. Their execution frameworks are based on Zheng et al. (2019). Yang et al. (2021) extracted events in a parallel mode, overcoming the dependence on argument role order. Huang and Jia (2021) and Zhu et al. (2022) took a different strategy. Huang and Jia (2021) exploited sentence community to determine the corresponding relation of entity-event, and this was done with a maximal clique composed of pseudo-triggers in Zhu et al. (2022).

However, these methods are under pipelined patterns and suffer from serial predictions, leading to error propagation. Therefore, this paper aims to develop a joint extraction model for document-level multi-event and argument cross-sentence.

6 Conclusions

This paper designs a token-token bidirectional event completed graph (TT-BECG) with the relation *eType-Role*$_1$-*Role*$_2$ as the edge type, followed by an edge-enhanced joint document-level event extraction model. First, the sequence labeling method is employed to transform the recognition objects from entities to tokens, preventing entity extraction in advance. Then, according to the given corpus, the target TT-BECG and corresponding adjacency matrix are constructed, which can accurately reveal which tokens play specific argument roles in an event of a specific event type, and realize the task transforms from the document-level event extraction to the structure and edge type prediction of the complete graph. Finally, a model is explored to approximate the given target token-token adjacency matrix and obtain the predicted token-token adjacency matrix. By decoding the predicted matrix, all events and event records in a document can be extracted. Extensive experiments have been conducted on ChFinAnn and DuEE-Fin corpora, and the results demonstrated the effectiveness and robustness of our scheme. The experimental code can be accessed at https://github.com/hawisdom/EDEE.

Limitations

As the experimental datasets are Chinese and the word segmentation tool is employed, some parsing errors may exist. Also, the token-token matrix is built on all tokens in each document, resulting in a large-scale matrix and the reduction of model training. All these are the limitations of this paper. Nevertheless, if the corpus is English, the first limitation does not exist. Also, the spatio-temporal efficiency in Table 1 is acceptable. Importantly, the experimental results obtained in this paper are based on the limitation, which indicates that it is effective to implement our model according to the segmentation results by syntactic tools.

Ethics Statement

Event extraction is an essential task of information extraction in NLP. We do not see any significant ethical concerns. Our work easily adapts to new event types and corpora by providing document samples. Therefore, the expected usage of our work is to identify interesting event records from user textual input (e.g., document and sentence).

Acknowledgements

This research was partially supported by the National Natural Science Foundation of China (61972184, 62272205, 62272206, and 62076112), the Science & Technology Project of the Department of Education of Jiangxi Province (GJJ210531), the Natural Science Foundation of Jiangxi Province (20212ACB202002 and 20192BAB207017), and the Funding Program for Academic and Technical Leaders in Major Disciplines of Jiangxi Province (20213BCJL22041).

References

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations (EMNLP)*, pages 42–49.

Yubo Chen, Liheng Xu, Kang Liu, Daojian Zeng, and Jun Zhao. 2015. Event extraction via dynamic multi-pooling convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 167–176.

Yubo Chen, Hang Yang, Kang Liu, Jun Zhao, and Yantao Jia. 2018. Collective event detection via a hierarchical and bias tagging networks with gated multi-level attention mechanisms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1267–1276.

Jacob Devlin, Mingwei 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 (NAACL-HLT)*, pages 4171–4186.

Xinya Du and Claire Cardie. 2020. Document-level event role filler extraction using multi-granularity contextualized encoding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8010–8020.

Seth Ebner, Patrick Xia, Ryan Culkin, Kyle Rawlins, and Benjamin Van Durme. 2020. Multi-sentence argument linking. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 8057–8077.

Cuiyun Han, Jinchuan Zhang, Xinyu Li, Guojin Xu, Weihua Peng, and Zengfeng Zeng. 2022. Duee-fin: A large-scale dataset for document-level event extraction. In *Proceedings of the CCF International Conference on Natural Language Processing and Chinese Computing*, pages 172–183.

Yusheng Huang and Weijia Jia. 2021. Exploring sentence community for document-level event extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 340–351.

Sha Li, Heng Ji, and Jiawei Han. 2021. Document-level event argument extraction by conditional generation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 894–908.

Yuan Liang, Zhuoxuan Jiang, Di Yin, and Bo Ren. 2022. RAAT: Relation-augmented attention transformer for relation modeling in document-level event extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 4985–4997.

Jiaju Lin, Qin Chen, Jie Zhou, Jian Jin, and Liang He. 2022. CUP: Curriculum learning based prompt tuning for implicit event argument extraction. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4245–4251.

Yaojie Lu, Hongyu Lin, Jin Xu, Xianpei Han, Jialong Tang, Annan Li, Le Sun, Meng Liao, and Shaoyi Chen. 2021. Text2Event: Controllable sequence-to-structure generation for end-to-end event extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 2795–2806.

Yubo Ma, Zehao Wang, Yixin Cao, Mukai Li, Meiqi Chen, Kun Wang, and Jing Shao. 2022. Prompt for extraction? PAIE: Prompting argument interaction for event argument extraction. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 6759–6774.

Thien Huu Nguyen, Kyunghyun Cho, and Ralph Grishman. 2016. Joint event extraction via recurrent neural networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pages 300–309.

Lei Sha, Feng Qian, Baobao Chang, and Zhifang Sui. 2018. Jointly extracting event triggers and arguments by dependency-bridge rnn and tensor-based argument interaction. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 5916–5923.

Qizhi Wan, Changxuan Wan, Rong Hu, and Dexi Liu. 2021. Chinese financial event extraction based on syntactic and semantic dependency parsing. *Chinese Journal of Computer*, 44(3):508–530.

Qizhi Wan, Changxuan Wan, Keli Xiao, Rong Hu, and Dexi Liu. 2023a. A multi-channel hierarchical graph attention network for open event extraction. *ACM Transactions on Information Systems (TOIS)*, 41(1):1–27.

Qizhi Wan, Changxuan Wan, Keli Xiao, Rong Hu, Dexi Liu, and Xiping Liu. 2023b. CFERE: Multi-type Chinese financial event relation extraction. *Information Sciences*, 630:119–134.

Qizhi Wan, Changxuan Wan, Keli Xiao, Dexi Liu, Qing Liu, Jiangling Deng, Wenkang Luo, and Rong Hu. 2022. Construction of a chinese corpus for multi-type economic event relation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21(6):1–20.

Runxin Xu, Tianyu Liu, Lei Li, and Baobao Chang. 2021. Document-level event extraction via heterogeneous graph-based interaction model with a tracker. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 3533–3546.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level chinese financial event extraction system based on automatically labeled training data. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics-System Demonstrations (ACL)*, pages 1–6.

Hang Yang, Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Taifeng Wang. 2021. Document-level event extraction via parallel prediction networks. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, pages 6298–6308.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5284–5294.

Zhisong Zhang, Xiang Kong, Zhengzhong Liu, Xuezhe Ma, and Eduard Hovy. 2020. A two-step approach for implicit event argument detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 7479–7485.

Shun Zheng, Wei Cao, Wei Xu, and Jiang Bian. 2019. Doc2EDAG: An end-to-end document-level framework for chinese financial event extraction. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 337–346.

Tong Zhu, Xiaoye Qu, Wenliang Chen, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Min Zhang. 2022. Efficient document-level event extraction via pseudo-trigger-aware pruned complete graph. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI)*, pages 4552–4558.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations.

□ A2. Did you discuss any potential risks of your work?
Not applicable. This does not apply.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Section Abstract and Introduction.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Left blank.

☑ B1. Did you cite the creators of artifacts you used?
Section 3.1.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 3.1.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 3.1.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section 3.1.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3.1.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 3.1.

C ☑ Did you run computational experiments?

Left blank.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Section 3.3.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 3.2 reports the experimental hyperparameters, but not shows the search process. Because the hyperparameters in this paper are set according to the hyperparameters commonly used in most existing models, and no hyperparameters are adjusted when model training.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 3.3.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 3.2.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

□ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

□ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

□ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

□ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

□ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.