Towards Better Graph-based Cross-document Relation Extraction via Non-bridge Entity Enhancement and Prediction Debiasing

Hao Yue[1,2], Shaopeng Lai[1,2], Chengyi Yang[1,2], Liang Zhang[1,2], Junfeng Yao[1,2], Jinsong Su[1,2*]
[1]School of Informatics, Xiamen University
[2]Xiamen Key Laboratory of Intelligent Storage and Computing, School of Informatics, Xiamen University
{yuehao,lzhang}@stu.xmu.edu.cn, laishaopeng.lsp@alibaba-inc.com,
yangchengyi031@gmail.com, {yao0010,jssu}@xmu.edu.cn

Abstract

Cross-document Relation Extraction aims to predict the relation between target entities located in different documents. In this regard, the dominant models commonly retain useful information for relation prediction via bridge entities, which allows the model to elaborately capture the intrinsic interdependence between target entities. However, these studies ignore the non-bridge entities, each of which co-occurs with only one target entity and offers the semantic association between target entities for relation prediction. Besides, the commonly-used dataset–CodRED contains substantial NA instances, leading to the prediction bias during inference. To address these issues, in this paper, we propose a novel graph-based cross-document RE model with non-bridge entity enhancement and prediction debiasing. Specifically, we use a unified entity graph to integrate numerous non-bridge entities with target entities and bridge entities, modeling various associations between them, and then use a graph recurrent network to encode this graph. Finally, we introduce a novel debiasing strategy to calibrate the original prediction distribution. Experimental results on the closed and open settings show that our model significantly outperforms all baselines, including the GPT-3.5-turbo and InstructUIE, achieving state-of-the-art performance. Particularly, our model obtains 66.23% and 55.87% AUC points in the official leaderboard[1] under the two settings, respectively, ranking the first place in all submissions since December 2023. Our code is available at https://github.com/DeepLearnXMU/CoRE-NEPD.

1 Introduction

Relation Extraction (RE) is a fundamental natural language processing (NLP) task, which aims to predict the relationship between two entities in a given context. Usually, conventional RE studies limit the context within a single sentence or document (Zeng et al., 2014; Santos et al., 2015; Cai et al., 2016; Zhang et al., 2018; Wang et al., 2022). However, since a large number of relational facts are not described in the same document (Yao et al., 2021), many researchers have begun to concentrate on cross-document RE (CoRE), where the given target entities (*head entity* and *tail entity*) are located in different documents (Yao et al., 2021; Wang et al., 2022; Wu et al., 2023).

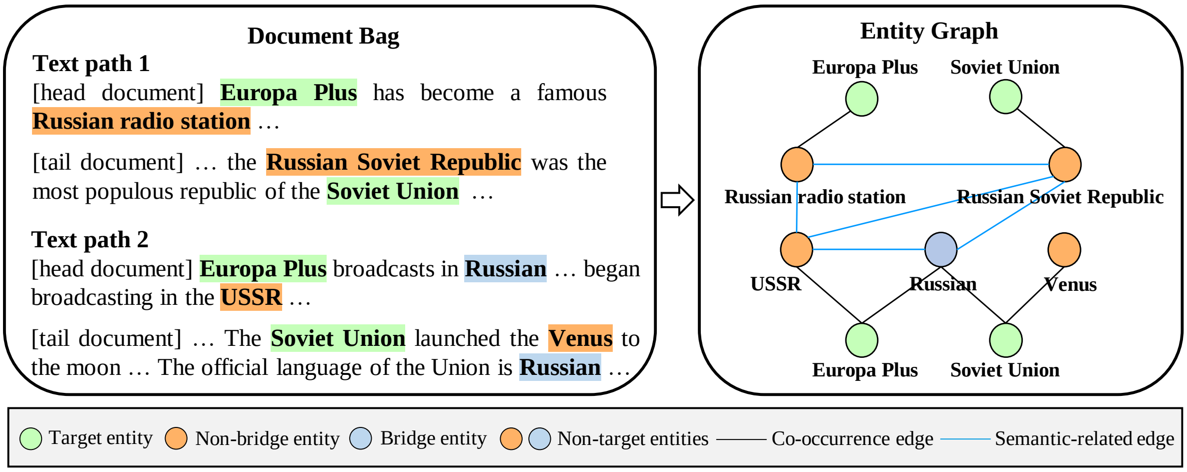
In this regard, Yao et al. (2021) first explore this task. They not only release the dataset CodRED, where relevant documents of target entities connected by bridge entities are organized as text paths, but also propose BERT-based models for this task. However, their models suffer from the negative effect of irrelevant context in the model input and do not fully leverage the connections across text paths. To solve these issues, Wang et al. (2022) propose an Entity-based Cross-path Relation Inference Method (Ecrim). They employ an entity-centered noise filter to refine the model input and incorporate a cross-path entity relation attention to capture the connections across different text paths. Unlike the Ecrim, Wu et al. (2023) present a local-to-global causal reasoning model (LGCR), which is a graph-based model using a local causality estimation algorithm to filter the noisy information. Despite their success, their models still suffer from two issues.

First, previous methods only consider the target entities and bridge entities, while ignoring the *non-bridge entities* that solely co-occur with only one target entity. According to the statistics of the CodRED dataset (Yao et al., 2021), we observe that on average each text path contains 18.8 non-target entities: 2.6 bridge entities while 16.2 non-bridge entities that may also help the relation prediction. For example, in the text path 1 of Figure 1, the target entities "*Europa Plus*" and "*Soviet Union*"

*Corresponding author.

[1] https://codalab.lisn.upsaclay.fr/competitions/3770#results

Findings of the Association for Computational Linguistics: ACL 2024, pages 680–691
August 11-16, 2024 ©2024 Association for Computational Linguistics



Document Bag

Text path 1

[head document] **Europa Plus** has become a famous **Russian radio station** …

[tail document] … the **Russian Soviet Republic** was the most populous republic of the **Soviet Union** …

Text path 2

[head document] **Europa Plus** broadcasts in **Russian** … began broadcasting in the **USSR** …

[tail document] … The **Soviet Union** launched the **Venus** to the moon … The official language of the Union is **Russian** …

Figure 1: A document bag containing two text paths and its corresponding entity graph. Note that we use different nodes to represent the same target entity for different text paths, which facilitates learning the relation between target entities specific to the text path.

only occur with the non-bridge entities "*Russian radio station*" and "*Russian Soviet Republic*", respectively. Thus, it is difficult to correctly predict their relation due to the absence of bridge entities. If non-target entities "*Russian radio station*" and "*Russian Soviet Republic*" are semantically related, then there may also be some relationship between the target entities "*Europa Plus*" and "*Soviet Union*". Therefore, intuitively, such semantic correlation can be exploited to benefit the relation prediction between target entities. **Second**, 85% of document bags are labeled as NA relation in the CodRED training set. As a result, the training dataset can only offer limited supervision signals for the model to learn non-NA relations, which leads to prediction bias.

To address these issues, in this paper, we propose a novel graph-based CoRE model with non-bridge entity enhancement and prediction debiasing. As illustrated in the right part of Figure 1, we represent each input document bag with a unified entity graph, where entity nodes are initialized with BERT representations. In this graph, each node indicates a target entity, bridge entity, or non-bridge entity, and two types of edges are introduced: 1) *co-occurrence edges*, each of which connects a target entity and a non-target entity that co-occurs with it. 2) *semantic-related edges*, which are used to connect any two semantic-related non-target entities. Then, we utilize Graph Recurrent Network (GRN) (Zhang et al., 2018) to encode this graph, where the interdependency among connected nodes is captured based on the recurrent gating mechanism. Afterward, we aggregate the entity representations learned from GRN and utilize a cross-path entity relation attention module to capture the connections across text paths.

Besides, inspired by the studies on other NLP tasks (Utama et al., 2020; Xiong et al., 2021; Wang et al., 2023a), we propose a simple yet effective prediction debiasing strategy to calibrate the relation prediction. Specifically, we introduce two prediction distributions: 1) $\overline{y}_{rela}$. To obtain this distribution, we fix the parameters of the original model and retrain a new classifier only using non-NA instances. Notably, the new classifier can avoid the negative impact of excessive NA instances and achieve better prediction performance on non-NA instances. 2) $\overline{y}_{bias}$. We mask the most important non-target entities of our entity graph as input to derive this distribution. Compared with the original prediction distribution, this distribution is more biased and thus can be used for debiasing in a manner of subtraction. Finally, we integrate the two newly-introduced prediction distributions to calibrate the original ones. To the best of our knowledge, our work is the first attempt to study the prediction debiasing in this task.

To investigate the effectiveness of our model, we conduct extensive experiments on both the closed and open settings of CodRED. Experimental results and in-depth analysis show that our model significantly outperforms all competitive baselines, including the LLMs. Particularly, compared with all submitted results since December 2023, our model ranks the first place in the official leaderboard.

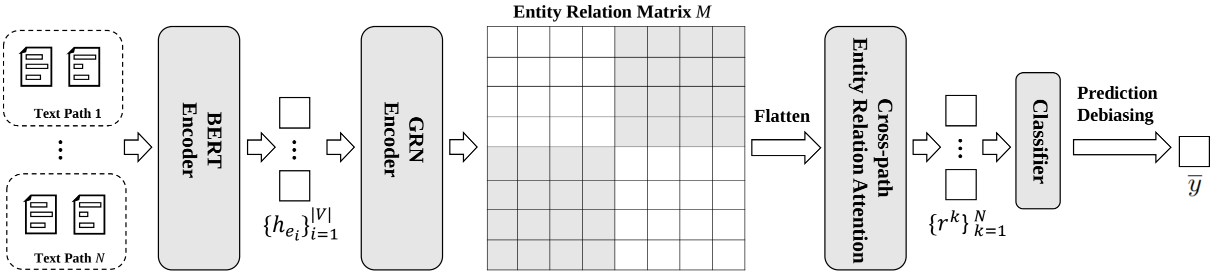


Figure 2: The overall architecture of our model. h_{e_i} is the entity representation after BERT encoding, $|V|$ is the number of entities nodes, M is the entity relation matrix, r^k is the final relation representation between target entities, $\overline{y}$ is the prediction distribution after debiasing

2 Our Model

In this section, we introduce our proposed model in detail. As shown in Figure 2, our model contains three important components: a BERT encoder, a GRN encoder, and a classifier with a cross-path entity relation attention module, and then we detail the prediction debiasing strategy used in inference.

2.1 BERT Encoder

Given a target entity pair, we first obtain relevant documents and conduct data preprocessing[2] to filter the noisy information over multiple documents. As a common practice, we use BERT[3] to learn the representations of entities, which will provide useful initial information for the subsequent GRN encoding. Concretely, for each entity e_i, we first insert a special symbol "*" at the start and end of entity mentions and obtain the mention representations by max-pooling operation. Then we collect all the mentions in a text path and follow Robin et al. (2019) to obtain its path-level representation h_{e_i}, using $h_{e_i} = \log \sum_{j=1}^{N_{e_i}} \exp\left(h_{m_j^i}\right)$, where N_{e_i} denotes the mention number of e_i in the text path, and m_j^i is the representation of its j-th entity mention.

2.2 GRN Encoder

To construct our GRN encoder, we first represent the whole input document bag as a unified entity graph, which facilitates the introduction of non-bridge entities to strengthen the semantic association between target entities. Then, we introduce a GRN to encode the graph, where the learned entity representations will provide information for the subsequent relation prediction.

2.2.1 Entity Graph

To facilitate the subsequent description, we take the document bag shown in Figure 1 as an example and describe how to use a unified entity graph to represent an undirected one $G{=}(V, E)$.

In the node set V, each node is a target entity, a bridge entity, or a non-bridge entity. Apparently, our entity graph considers more information than previous studies. Let us revisit the example in Figure 1, where we first identify the bridge entity "*Russian*", the non-bridge entities "*Russian radio station*" and "*Russian Soviet Republic*", and then include them with the target entities "*Europa Plus*" and "*Soviet Union*" into the entity graph. Note that we use different nodes to represent the same head entity for different text paths, which facilitates learning the relation between target entities specific to the text path.

To capture various kinds of interdependences between entities for cross-document RE, we consider two kinds of edges in the edge set E: (1) each target entity and each non-target entity (bridge or non-bridge entities) within the same document are connected via a *co-occurrence edge*; (2) any two semantic-related non-target entities are connected by a *semantic-related edge*. During this process, we calculate the cosine similarity between the representations of any two non-target entities, and determine that they are semantically related if their similarity is greater than a threshold η. Back to Figure 1, "*Europa Plus*" and "*Russian radio station*" are connected by a co-occurrence edge, and "*Russian radio station*" and "*Russian Soviet Republic*" are connected by a semantic-related edge.

2.2.2 Encoding with GRN

We then introduce GRN (Zhang et al., 2018) to encode this graph. Typically, it updates node representations using recurrent gating mechanisms,

[2] The detailed data preprocessing we used can be found in Appendix A

[3] We analyze the effect of the BERT version in Appendix C

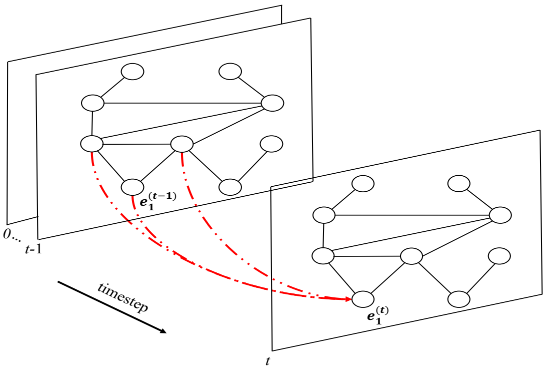


Figure 3: An example of message-passing procedures of our GRN encoder at the t-th timestep. The update of $\boldsymbol{e}_1^{(t)}$ is in the red dash lines.

and thus has been widely used in many NLP tasks (Yin et al., 2019, 2020; Lai et al., 2021). Here, we choose GRU to update node representations since it has fewer parameters and better efficiency.

In Figure 3, we show the procedure of updating node representations in our graph at the t-th timestep. Specifically, for the node of entity e_i, we first gather the information from its connected entity nodes of the graph:

$$\boldsymbol{c}_i^{(t)} = \sum_{j \in A(e_i)} \boldsymbol{e}_j^{(t-1)}, \tag{1}$$

where $\boldsymbol{c}_i^{(t)}$ is the collected context information used to update the node representation of entity e_i, $A(e_i)$ represents the set of neighboring nodes of entity e_i, $\boldsymbol{e}_j^{(t-1)}$ is the node representation of entity e_j at the $(t-1)$-th timestep. Besides, we initialize the entity nodes with their representations learned from BERT encoder: $\boldsymbol{e}_i^{(0)} = h_{e_i}$, where $1 \leq i \leq |V|$. Afterward, we update the node representation of entity e_i in the following way:

$$\boldsymbol{r}_i^{(t)} = \sigma(W^r \boldsymbol{c}_i^{(t)} + U^r \boldsymbol{e}_i^{(t-1)}), \tag{2}$$

$$\boldsymbol{z}_i^{(t)} = \sigma(W^z \boldsymbol{c}_i^{(t)} + U^z \boldsymbol{e}_i^{(t-1)}), \tag{3}$$

$$\boldsymbol{u}_i^{(t)} = \tanh(W^u \boldsymbol{c}_i^{(t)} + U^u (\boldsymbol{r}_i^{(t)} \odot \boldsymbol{e}_i^{(t-1)})), \tag{4}$$

$$\boldsymbol{e}_i^{(t)} = (1 - \boldsymbol{z}_i^{(t)}) \odot \boldsymbol{u}_i^{(t)} + \boldsymbol{z}_i^{(t)} \odot \boldsymbol{e}_i^{(t-1)}, \tag{5}$$

where $\boldsymbol{r}_i^{(t)}$ and $\boldsymbol{z}_i^{(t)}$ are reset and update gates, $\odot$ is the Hadamard product, $\boldsymbol{u}_i^{(t)}$ denotes the temporary node representations of $\boldsymbol{e}_i$ at t-th step, W^* and U^* are trainable parameter matrixes.

2.3 Classifier with Cross-path Entity Relation Attention

Following Wang et al. (2022), on the basis of the entity representations learned from GRN, we utilize a cross-path entity relation attention to model the connections across different text paths, aggregating better relation representations of the target entity pair. Then, we feed these aggregated representations into an MLP classifier for prediction.

Specifically, we first collect all the entity representations learned from the GRN encoder and then construct a entity relation matrix $M \in \mathbb{R}^{|V| \times |V|}$, with each element $r_{i,j}$ denoting the representation of the relation between the entities e_i and e_j:

$$r_{i,j} = \mathrm{ReLU}(W(\mathrm{ReLU}(W^u \boldsymbol{e}_i + W^v \boldsymbol{e}_j)), \tag{6}$$

where $\boldsymbol{e}_i$, $\boldsymbol{e}_j$ are the representations of entity e_i, e_j respectively, W^* are trainable parameter matrixes.

Then we flatten the entity relation matrix M and perform self-attention on it, to capture the intra- and inter-path dependencies. By doing so, we can obtain the representation r^k of the relation between the target entities of each text path k. Subsequently, we feed r^k into the MLP classifier for prediction:

$$\overline{y}^k = \mathrm{MLP}(r^k), \tag{7}$$

where $\overline{y}^k$ is the predicted relation distribution for text path k.

Finally, to obtain the bag-level prediction $\overline{y}$, we follow Wang et al. (2022) to perform a max-pooling operation as follows:

$$\overline{y} = \mathrm{Max}(\{\overline{y}^k\}_{k=1}^N), \tag{8}$$

where N denotes the number of text paths in a document bag.

2.4 Model Inference with Prediction Debiasing

Following common practice, we train our model by minimizing the cross-entropy loss of training data. However, as previous analysis, the training data contains numerous target entity pairs with NA relation, which leads to the prediction biasing during inference. To address this issue, we propose a simple yet effective debiasing strategy that introduces two prediction distributions to calibrate the original prediction ones in the following way:

$$\overline{y} = \overline{y} + \lambda\ (\overline{y}_{rela} - \overline{y}_{bias}), \tag{9}$$

where λ is a hyper-parameter controlling the effects of newly-introduced prediction distributions. Here, we give detailed descriptions to $\overline{y}_{rela}$ and $\overline{y}_{bias}$:

- $\overline{y}_{rela}$. This prediction distribution is used to refine the prediction of the model on the in-

	Closed			Open	
	Train	**Dev**	**Test**	**Dev**	**Test**
#Bags(non-NA)	2,733	1,010	1,012	1,010	5,523
#Bags(NA)	16,668	4,558	4,523	4,558	
#Text paths(non-NA)	8,263	2,558	40,524	15,072	7,7840
#Text paths(NA)	120,925	38,182		62,863	
#Tokens/Doc	4,938.6	5,031.6	5,129.2	5,934.4	5,983.4
#Path/Bag	6.67	7.31	7.32	13.99	14.09

Table 1: Statistics of CodRED. Note that, Bags(non-NA) denotes the bags with non-NA relations, and Bags(NA) denotes the bags with NA relations.

stances with non-NA relations. To obtain this distribution, we stack a new classifier on the original model. During the training process, we fix all parameters of the original model and only tune this classifier using the non-NA instances. Notably, unlike the original classifier, the training of this classifier avoids the negative impact of excessive NA instances. Thus, it can achieve better prediction performance on non-NA instances.

- $\overline{y}_{bias}$. Compared with the original prediction distribution, this distribution is more biased and thus can be used to debias the original prediction distribution in a manner of subtraction. Notably, unlike the above $\overline{y}_{rela}$, we do not retrain the model or classifier to obtain $\overline{y}_{bias}$. To obtain this distribution, we first quantify the importance of each non-target entity with the average weight of target entities attending to the non-target entity. Subsequently, we mask the most important 50% non-target entities and then feed the remaining sub-graph into the GRN to derive $\overline{y}_{bias}$. Apparently, due to this sub-graph lacking some important non-target entities, $\overline{y}_{bias}$ prefers NA relation and thus is more biased than $\overline{y}$.

3 Experiments

3.1 Setup

Dataset. To evaluate our model, we use the commonly-used dataset–CodRED (Yao et al., 2021) to conduct experiments under two settings: 1) **closed setting**, where the related documents of target entities are given in advance for constructing the text path. 2) **open setting**, where we have to first retrieve related documents from Wikipedia and then evaluate the model performance. Table 1 shows the detailed statistics of CodRED. Note that the training data of CodRED contains 16,668 document bags with NA relation and 2,733 document bags with non-NA relations, where the significant number difference between NA instances and non-NA instances leads to the prediction bias.

Settings. When constructing our model, we set the similarity threshold η for semantic-related edges to 0.6.[4] As for the debiasing strategy, we set the mask rate of important non-target entities to 0.5 and λ to 0.1, which will be analyzed in Section 3.3. Besides, we employ a 3-layer GRN to encode our unified entity graph and a 2-layer Transformer for cross-path entity relation attention, where the embedding size and hidden state dimension are both set to 768. To effectively train our model, we employ AdamW with a learning rate of 3e-5. To ensure a fair comparison, we follow Yao et al. (2021) to train the model on the closed setting, where extra document-level data are involved.[5]

As implemented in previous studies (Yao et al., 2021; Wang et al., 2022), we use four evaluation metrics for the development set: F1, AUC, P@500, and P@1000, and two evaluation metrics for the test set: F1 and AUC. Additionally, we compare our model with the LLMs using micro-F1. Finally, following Yao et al. (2021), we obtain the evaluation scores on the test set by submitting prediction results into Codalab.[6]

3.2 Baselines

We compare our model with the following baselines:

- **End-to-end** (Yao et al., 2021). This model employs BERT to obtain the representations of text paths. Then, a selective attention module is used to obtain the aggregated representations of target entities. Finally, the aggregated representations are fed into a classifier to predict the relation between target entities.
- **Ecrim** (Wang et al., 2022). It is our most important baseline. This model first uses an entity-based document-context filter to retain useful information in the given documents by using the bridge entities in the text paths. Then, it is equipped with a cross-path entity relation attention, which allows entity relations across text paths to interact with each other.
- **LGCR** (Wu et al., 2023). This model first estimates the causal effect for each semantic

[4] We analyze the effect of η in Appendix B

[5] We investigate the effect of extra document-level data in Section 3.7

[6] https://codalab.lisn.upsaclay.fr/competitions/3770

Model	Closed						Open					
	Dev				Test		Dev				Test	
	F1	AUC	P@500	P@1000	F1	AUC	F1	AUC	P@500	P@1000	F1	AUC
End-to-end (Yao et al., 2021)†	51.26	47.94	62.80	51.00	51.02	47.46	47.23	40.86	59.00	46.30	45.06	39.05
Ecrim (Wang et al., 2022)†	61.12	60.91	**78.89**	60.17	62.48	60.67	–	–	–	–	–	–
Ecrim (Wang et al., 2022)	61.42	61.05	78.04	60.44	62.73	60.84	51.28	49.65	69.25	51.55	51.78	49.58
LGCR (Wu et al., 2023)†	61.67	63.17	76.65	61.84	61.08	60.75	52.96	51.48	**70.06**	52.19	53.45	50.15
LGCR (Wu et al., 2023)	61.72	63.05	76.83	61.97	61.25	60.44	53.02	51.26	69.67	52.25	53.60	50.12
Ours	**63.63**	**65.01**	77.84	**64.03**	**64.41**	**66.23**	**54.49**	**54.92**	68.66	**53.84**	**56.68**	**55.87**

Table 2: Experimental results on the CodRED dataset. † indicates previously reported scores.

unit (text path, head entity, tail entity, and bridge entity). Then, it constructs a global reasoning graph based on the co-occurrence of entity mentions and the structure of text paths, and uses the relative causal association calculated by local causal effect to control the message propagation ability between nodes.

The number of trainable parameters for our model and the above baselines are 1.30×10^8, 1.08×10^8, 1.23×10^8, and 1.19×10^8, respectively.

To further verify the performance of our model, we compare it with powerful LLMs. In addition to the commonly-used **GPT-3.5-turbo** (Ouyang et al., 2022), we consider a variant of InstructUIE (Wang et al., 2023b), termed as **InstructUIE-FT**, which is fine-tuned on CodRED to enhance its cross-document RE ability. Note that InstructUIE is a strong information extraction framework that captures the inter-task dependency to uniformly model various information extraction tasks.

To investigate the few-shot ability of LLMs in classification tasks, the standard practice based on LLMs introduces an exemplar for each candidate label to investigate the few-shot ability of LLMs (Yoo et al., 2022; Dong et al., 2023; Fan et al., 2023). However, in the CoRE task, the number of candidate labels is 276, making it impractical to introduce an exemplar for each label due to the maximum context length of the LLMs. Therefore, we only test the zero-shot performance of LLMs. The detailed settings for these LLMs are shown in Appendix D.

3.3 Effect of the Adaptive Parameter λ

We first investigate the impact of the hyper-parameter λ (See Equation 9) on the development set under the closed setting. To this end, we gradually vary λ from 0.05 to 0.25 with an increment of 0.05 in each step.

As shown in Figure 4, we find that our model achieves the best performance when λ=0.1. Therefore, we set λ=0.1 for all experiments thereafter.

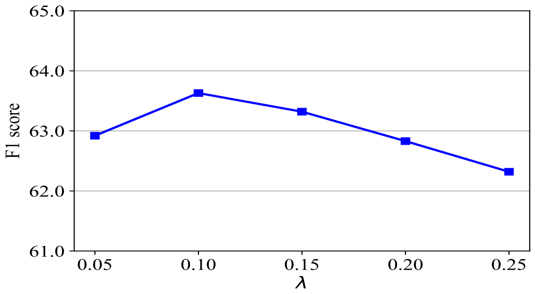


Figure 4: The performance of our model (F1 score) on the CodRED development set under the closed setting, using different λ .

Model	Closed-Dev	Open-Dev
	micro-F1	micro-F1
End-to-end	78.24	77.63
Ecrim	82.59	81.08
LGCR	82.07	81.79
GPT-3.5-turbo	28.05	25.31
InstructUIE	70.34	64.05
InstructUIE-FT	80.72	79.66
Ours	**84.35**	**83.92**

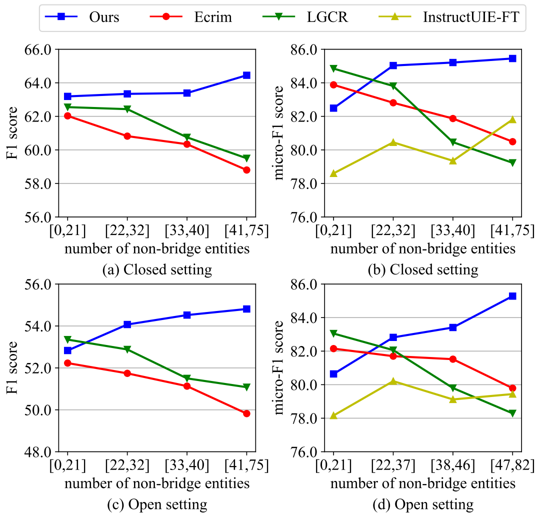
Table 3: Experimental results with LLMs.

3.4 Main Results

Table 2 shows the experimental results under two settings. Note that the performance of our reproduced LGCR and Ecrim model are comparable to that of their original paper, proving that our experimental comparison is convincing. Overall, under both settings, our model exhibits better performance in most metrics than all baselines, except for P@500. Most importantly, we submit the test results to the official competition leaderboard, where our model obtains 66.23% and 55.87% AUC points under the closed and open settings, respectively, ranking the first place in all submitted results since December 2023.

Model	Closed						Open					
	Dev				Test		Dev				Test	
	F1	**AUC**	**P@500**	**P@1000**	**F1**	**AUC**	**F1**	**AUC**	**P@500**	**P@1000**	**F1**	**AUC**
Ours	63.63	65.01	77.84	64.03	64.41	66.23	54.49	54.92	68.66	53.84	56.68	55.87
w/o $\overline{y}_{rela}$	63.12	63.77	77.54	63.43	63.30	64.69	53.74	53.31	67.06	52.94	55.01	54.95
w/o $\overline{y}_{bias}$	63.32	64.85	77.44	63.63	63.85	65.78	54.07	54.16	67.66	53.04	55.66	55.38
w/o NBE	63.03	64.36	77.42	63.31	63.81	65.28	53.85	54.02	67.24	52.87	55.51	55.14
w/o GRN	62.88	63.79	77.35	62.89	63.15	64.87	53.33	53.10	67.02	52.53	54.89	54.44
w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$	61.88	63.12	77.40	62.23	62.23	62.77	52.26	52.45	66.86	52.14	53.25	52.85
w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$, NBE	61.15	62.36	77.04	61.42	61.62	61.87	51.49	51.14	66.07	51.45	52.12	52.08
w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$, SRE	61.27	62.85	77.15	61.68	62.01	62.29	51.82	51.95	66.49	51.76	52.88	52.32

Table 4: Ablation study of our model on the CodRED dataset. Note that **NBE** refers to the non-bridge entities and **SRE** refers to the semantic-related edges in the graph.



(a) Closed setting

(b) Closed setting

(c) Open setting

(d) Open setting

Figure 5: The model performance on different development subsets of CodRED under two settings.

Moreover, we compare our model with LLMs in Table 3. Under both settings, we can find that our model significantly outperforms the LLMs, including InstructUIE-FT that has also been fine-tuned on CodRED. These results once again demonstrate the effectiveness of our model.

3.5 Impact of Non-bridge Entity Number

To assess the impact of non-bridge entities on our model, we first sort the development set according to the number of non-bridge entities in ascending order, and then equally split it into four subsets. After that, we compare the model performance on these subsets. To clearly display the experimental results, we only compare ours with Ecrim, LGCR and InstructUIE-FT, which are competitive baselines according to the results reported in Table 3.

As shown in Figure 5, regardless of the setting, as the number of non-bridge entities increases, the performance advantage of our model becomes increasingly apparent. These results strongly demonstrate the generality and effectiveness of our model.

3.6 Ablation Study

To investigate the effectiveness of different components of our model, we further compare our model with the following variants in Table 4:

- *w/o* $\overline{y}_{rela}$. In this variant, we only use $\overline{y}_{bias}$ for prediction debiasing.
- *w/o* $\overline{y}_{bias}$. We only use $\overline{y}_{rela}$ for prediction debiasing in this variant.
- *w/o* $\overline{y}_{rela}$, $\overline{y}_{bias}$. In this variant, we do not calibrate the model prediction.
- *w/o NBE.* In this variant, we remove all non-bridge entities from our entity graph.
- *w/o GRN.* In this variant, we remove the GRN Encoder from our model.
- *w/o* $\overline{y}_{rela}$, $\overline{y}_{bias}$, *NBE(Non-bridge entities).* In this variant, we directly remove all non-bridge entities from our entity graph. In other words, we only consider target entities and bridge entities in this variant. Notice that the main difference between this variant and Ecrim is that it adds an additional GRN on the top of BERT encoder.
- *w/o* $\overline{y}_{rela}$, $\overline{y}_{bias}$, *SRE(Semantic-related edges).* When constructing this variant, we remove the debiasing module and all semantic-related edges from our entity graph.

Table 4 lists the experimental results, where, all variants are inferior to our model, verifying the effectiveness of each proposed module.

3.7 Effect of Extra Document-level Data

To investigate the effect of extra document-level data in training, we only use the cross-document data to train various models.

Model	Closed-set				Open-set			
	Dev		Test		Dev		Test	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
End-to-end (Yao et al., 2021)†	26.56	15.67	–	–	22.06	11.43	–	–
Ecrim (Wang et al., 2022)	39.19	29.85	36.41	27.40	25.01	18.04	24.93	18.97
LGCR (Wu et al., 2023)	40.73	32.81	36.67	28.01	27.54	23.57	26.85	23.32
Ours	**42.26**	**34.13**	**38.22**	**33.15**	**30.26**	**25.54**	**29.05**	**26.64**

Table 5: Experimental results with cross-document-only supervision on the Codred dataset. † indicates previously reported scores.

From Table 5, we observe that the performance of various models on the CoRE task sharply decreases, which demonstrates that the extra document-level data can help models to capture more useful information. Moreover, we can observe that our model still significantly outperforms all baselines in this setting, confirming the capability of our model.

3.8 Case Study

Table 9 in the Appendix displays the prediction results of different models on two text paths sampled from the test set. In text path 1, due to the lack of bridge entities, Ecrim is unable to connect the head and tail entities, leading to the incorrect relation prediction. By contrast, our model leverages substantial non-bridge entities of text path 1 to build connections between target entities. Thus, both Ours and w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$ successfully predict the relation of text path 1, demonstrating that non-bridge entities can indeed provide useful supplementary information for the cross-document RE. Meanwhile, in text path 2, both Ecrim and w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$ mistakenly predict the relation between target entities as NA. Only when the prediction debiasing strategy is used, our model can calibrate the prediction, thereby correctly predicting the relation as continent. This is consistent with the results reported in Table 4, further verifying the effectiveness of our strategy.

4 Related Work

Most studies on RE mainly focus on the sentence-level RE (Cai et al., 2016; Zhang et al., 2018, 2019; Ikuya et al., 2020; Zhang et al., 2023b) and document-level RE (Zeng et al., 2020; Wang et al., 2021; Zhang et al., 2022, 2023a), which are committed to identifying the relation between target entities from a sentence and document, respectively. Unlike these studies, in this work, we concentrate on cross-document RE that aims at predicting the relation between target entities located in different documents. Yao et al. (2021) comprehensively investigated this task and released the first human-annotated CoRE dataset, CodRED. Besides, they explore an end-to-end model that jointly considers documents in text paths to predict the relation. However, it not only suffers from the negative effect of irrelevant context in text paths but also does not fully leverage the interconnections across text paths. To address the above-mentioned issues, Wang et al. (2022) propose Entity-based Cross-path Relation InferenceMethod (Ecrim). Typically, it uses an attention mechanism to capture the connection among different text paths through bridge entities, which is important to predict the final relation. Recently, Wu et al. (2023) put forward local-to-global causal reasoning (LGCR). To aggregate information over multiple text paths, they construct a global heterogeneous graph, where a local causality estimation algorithm is proposed to assess the importance of different nodes in the graph.

However, their methods suffer from two limitations: 1) ignore the non-bridge entities, which exist broadly in each text path and can offer semantic associations between target entities, especially in the absence of bridge entities. 2) ignore the bias caused by the prominent number difference between NA and non-NA instances. In this work, we propose a graph-based model to fully exploit non-bridge entities for cross-document RE. Besides, along the research line of debiasing in NLP (Joshua R et al., 2021; Xiong et al., 2021; Wang et al., 2023a), we propose a simple yet effective prediction debiasing strategy to refine the prediction of our model.

5 Conclusion and Future work

In this paper, we propose a novel graph-based cross-document RE model. Concretely, we first represent the input bag as a unified entity graph, where abundant non-bridge entities are introduced to provide useful information. Then, we use GRN to encode this graph, where the learned entity representations form the basis for subsequent relationship prediction. Besides, we propose a simple yet effective debiasing strategy to refine the original prediction distribution. To the best of our knowledge, the graph-based model with non-bridge entities and our debiasing strategy has not been explored before. Extensive experiments on the commonly-used dataset CodRED demonstrate the superiority of our model. In the future, we will study how to introduce more external knowledge to refine our model.

Limitations

One limitation of the present work lies in that we only rely on the attention score to measure the importance of nodes in the graph, and do not consider dynamically adjusting the importance score. Additionally, in the entity-based graph, the edges are connected using a heuristic method, which may overlook useful information.

Acknowledgements

The project was supported by National Natural Science Foundation of China (No. 62276219), and the Public Technology Service Platform Project of Xiamen (No. 3502Z20231043). We also thank the reviewers for their insightful comments.

References

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *ACL 2016*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. 2023. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Caoyun Fan, Jidong Tian, Yitian Li, Hao He, and Yaohui Jin. 2023. Comparable demonstrations are important in in-context learning: A novel perspective on demonstration selection. *arXiv preprint arXiv:2312.07476*.

Ridong Han, Tao Peng, Chaohao Yang, Benyou Wang, Lu Liu, and Xiang Wan. 2023. Is information extraction solved by chatgpt? an analysis of performance, evaluation criteria, robustness and errors. *arXiv preprint arXiv:2305.14450*.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Yamada Ikuya, Asai Akari, Shindo Hiroyuki, Takeda Hideaki, and Matsumoto Yuji. 2020. Luke: Deep contextualized entity representations with entityaware self-attention. In *EMNLP 2020*.

Minot Joshua R, Cheney Nicholas, Maier Marc, Elbers Danne C, Danforth Christopher M, and Dodds Peter Sheridan. 2021. Interpretable bias mitigation for textual data: Reducing gender bias in patient notes while maintaining classification performance. *arXiv preprint arXiv:2103.05841*.

Shaopeng Lai, Ante Wang, Fandong Meng, Jie Zhou, Yubin Ge, Jiali Zeng, Junfeng Yao, Degen Huang, and Jinsong Su. 2021. Improving graph-based sentence ordering with iteratively predicted pairwise orderings. In *EMNLP 2021*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*.

Jia Robin, Wong Cliff, and Poon Hoifung. 2019. Document-level n-ary relation extraction with multiscale representation learning. In *NAACL 2019*.

Cicero Nogueira dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *ACL 2015*.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. 2020. Towards debiasing NLU models from unknown biases. In *EMNLP 2020*.

Fei Wang, James Y. Huang, Tianyi Yan, Wenxuan Zhou, and Muhao Chen. 2023a. Robust natural language understanding with residual attention debiasing. In *ACL 2023 findings*.

Fengqi Wang, Fei Li, Hao Fei, Jingye Li, Shengqiong Wu, Fangfang Su, Wenxuan Shi, Donghong Ji, and Bo Cai. 2022. Entity-centered cross-document relation extraction. In *EMNLP 2022*.

Xiao Wang, Weikang Zhou, Can Zu, Han Xia, Tianze Chen, Yuansen Zhang, Rui Zheng, Junjie Ye, Qi Zhang, Tao Gui, Jihua Kang, Jingsheng Yang, Siyuan Li, and Chunsai Du. 2023b. Multi-task instruction tuning for unified information extraction. *arXiv preprint arXiv:2304.08085*.

Xu Wang, Chen Kehai, and Zhao Tiejun. 2021. Discriminative reasoning for document-level relation extraction. In *ACL 2020*.

Haoran Wu, Xiuyi Chen, Zefa Hu, Jing Shi, Shuang Xu, and Bo Xu. 2023. Local-to-global causal reasoning for cross-document relation extraction. *IEEE/CAA JOURNAL OF AUTOMATICA SINICA*, 10:1608–1621.

Ruibin Xiong, Yimeng Chen, Liang Pang, Xueqi Cheng, ZhiMing Ma, and Yanyan Lan. 2021. Uncertainty calibration for ensemble-based debiasing methods. In *NIPS 2021*.

Yuan Yao, Jiaju Du, Yankai Lin, Peng Li, Zhiyuan Liu, Jie Zhou, and Maosong Sun. 2021. Codred: A cross-document relation extraction dataset for acquiring knowledge in the wild. In *EMNLP 2021*.

Yongjing Yin, Shaopeng Lai, Linfeng Song, Chulun Zhou, Xianpei Han, Junfeng Yao, and Jinsong Su. 2020. An external knowledge enhanced graph-based neural network for sentence ordering. *Journal of Artificial Intelligence Research*.

Yongjing Yin, Linfeng Song, Jinsong Su, Jiali Zeng, Chulun Zhou, and Jiebo Luo. 2019. Graph-based neural sentence ordering. In *IJCAI 2019*.

Kang Min Yoo, Junyeob Kim, Hyuhng Joon Kim, Hyunsoo Cho, wiyeol Jo, Sang-Woo Lee, Sang-goo Lee, and Taeuk Kim. 2022. Ground-truth labels matter: A deeper look into input-label demonstrations. In *EMNLP 2022*.

Chenhan Yuan, Qianqian Xie, and Sophia Ananiadou. 2023. Zero-shot temporal relation extraction with chatgpt. *arXiv preprint arXiv:2304.05454*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. relation classification via convolutional deep neural network. In *ACL 2014*.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *EMNLP 2020*.

Liang Zhang, Jinsong Su, Yidong Chen, Zhongjian Miao, Zijun Min, Qingguo Hu, and Xiaodong Shi. 2022. Towards better document-level relation extraction via iterative inference. In *EMNLP 2022*.

Liang Zhang, Jinsong Su, Zijun Min, Zhongjian Miao, Qingguo Hu, Biao Fu, Xiaodong Shi, and Yidong Chen. 2023a. Exploring self-distillation based relational reasoning training for document-level relation extraction. In *AAAI 2023*.

Liang Zhang, Chulun Zhou, Fandong Meng, Jinsong Su, Yidong Chen, and Jie Zhou. 2023b. Hypernetwork-based decoupling to improve model generalization for few-shot relation extraction. In *EMNLP 2023*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP 2018*.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *ACL 2019*.

A Data preprocessing

Strictly following Wang et al. (2022), we preprocess our experimental datasets. As shown in Fig 7, we first retrieve relevant text paths for the target entities from the CodRED dataset, forming a document bag. Note that, since the length of each document in a text path may exceed the limit of BERT, we then use an entity-based document-context filter (Wang et al., 2022) to select salient sentences for each document and ensure the total length of a text path is less than 512. Finally, we concatenate the head document and tail document from each text path and then obtain the input of BERT, where the number of text paths in a document bag is the batch size.

B Effect of the threshold η for semantic-related edge

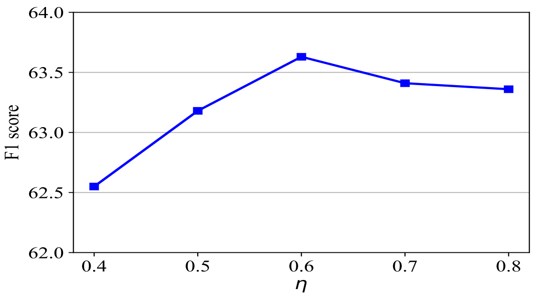


Figure 6: The performance of our model (F1 score) on the development set under the closed setting, using different η.

We also investigate the impact of the hyper-parameter η on the development set under the closed setting. To this end, we gradually vary η from 0.4 to 0.8 with an increment of 0.1 in each step. As shown in Figure 6, we find that our model achieves the best performance when η=0.6. Therefore, we set η=0.6 for all experiments.

C Effect of the BERT version

Model	Closed-Test		Open-Test	
	F1	AUC	F1	AUC
LGCR-BERT$_{base}$	61.08	60.75	53.45	50.15
LGCR-BERT$_{large}$	62.16	61.51	54.34	51.07
Ours-BERT$_{base}$	64.41	66.23	56.68	55.87
Ours-BERT$_{large}$	**65.27**	**66.98**	**57.44**	**56.43**

Table 6: Experimental results with previous SOTA model, using different BERT versions.

Model	Closed-Dev	Open-Dev
	micro-F1	**micro-F1**
LGCR-BERT$_{base}$	82.07	81.79
LGCR-BERT$_{large}$	83.12	82.65
GPT-3.5-turbo	28.05	25.31
InstructUIE	70.34	64.05
InstructUIE-FT	80.72	79.66
Ours-BERT$_{base}$	84.35	83.92
Ours-BERT$_{large}$	**85.18**	**84.60**

Table 7: Experimental results with LLMs, using different BERT versions.

	GPT-3.5-turbo	InstructUIE
Prompt 1	Given the list of relations: [“highway system”, “country”, “place of birth”, ...], read the given text path [text path] and predict the relation between head entity: [h] and tail entity: [t]. Answer in the format [“relation label”, “confidence score(Decimal between 0-1)”] without any explanation.	Text: [text path]. In the above text, what is the relationship between [head entity] and [tail entity]? Option: [“highway system”, “country”, “place of birth”, ...]. Answer:
Prompt 2	Analyze the information provided in the text path [text path], which includes mentions of the head entity [h] and the tail entity [t]. Infer the relation from the given relation set: [“highway system”, “country”, “place of birth”, ...] between the target entity pair. Please consider the entire path and answer in the format [“relation”, “confidence score(Decimal between 0-1)”] without any explanation.	Text: [text path]. Find the relationship between [head entity] and [tail entity] in the above text. Option: [“highway system”, “country”, “place of birth”, ...]. Answer:
Prompt 3	Given a text path [text path] containing mentions of the head entity [h] and the tail entity [t] in separate sentences, your goal is to identify and infer the relation from the pre-defined set [“highway system”, “country”, “place of birth”, ...] between the target entity pair. Pay close attention to the specific information provided in the path. Answer in the format [“relation”, “confidence score(Decimal between 0-1)”] without any explanation.	Text: [text path]. Given the above text, please tell me the relationship between [head entity] and [tail entity]. Option: [“highway system”, “country”, “place of birth”, ...]. Answer:

Table 8: Zero-shot prompts for GPT-3.5-turbo and InstructUIE on CoRE task

We also investigate the impact of the BERT version. As shown in Table 6 and Table 7, when using BERT$_{large}$, our model still significantly outperforms both the previous SOTA: LGCR and LLM-based methods: GPT-3.5-turbo and InstructUIE-FT, demonstrating the effectiveness of our model.

D Experimental settings for LLMs

The detailed experimental settings for the LLMs are as follows:

- GPT-3.5-turbo. As implemented in previous studies (Han et al., 2023; Wang et al., 2023b; Yuan et al., 2023), we devise three diverse prompts as shown in Table 8, which prompt LLM to generate both the relation label and the corresponding confidence scores. Subsequently, to obtain the bag-level relation prediction, we select the highest confidence score among all text paths within a document bag. We conduct experiment under each prompt separately and then average the results under all prompts as the final experimental result.
- InstructUIE and InstructUIE-FT. We first use LoRA (Hu et al., 2021) to obtain InstructUIE-FT. During this process, we set the learning rate to 5e-5 and the batch size to 8. As for the prompt, we follow Wang et al. (2023b) to design three prompts as shown in Table 8. Note that, to obtain the bag-level prediction, we first get the prediction score of the relation label generated by the model. After that, we take the relation label with the highest prediction score among all text paths as the final prediction of a document bag. Like GPT-3.5-turbo, we also average the results under all prompts as the final experimental result.

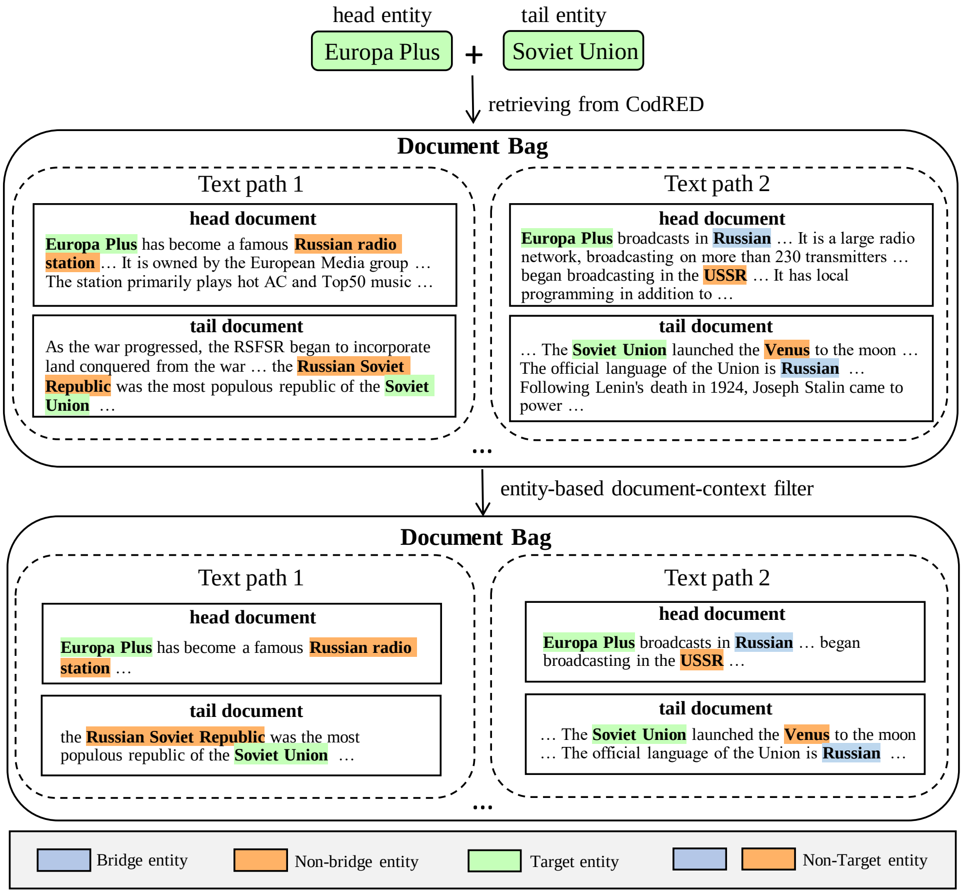

head entity
Europa Plus
+
tail entity
Soviet Union
retrieving from CodRED
Document Bag
Text path 1
head document
Europa Plus has become a famous Russian radio station … It is owned by the European Media group … The station primarily plays hot AC and Top50 music …
tail document
As the war progressed, the RSFSR began to incorporate land conquered from the war … the Russian Soviet Republic was the most populous republic of the Soviet Union …
Text path 2
head document
Europa Plus broadcasts in Russian … It is a large radio network, broadcasting on more than 230 transmitters … began broadcasting in the USSR … It has local programming in addition to …
tail document
… The Soviet Union launched the Venus to the moon … The official language of the Union is Russian … Following Lenin's death in 1924, Joseph Stalin came to power …
…
entity-based document-context filter
Document Bag
Text path 1
head document
Europa Plus has become a famous Russian radio station …
tail document
the Russian Soviet Republic was the most populous republic of the Soviet Union …
Text path 2
head document
Europa Plus broadcasts in Russian … began broadcasting in the USSR …
tail document
… The Soviet Union launched the Venus to the moon … The official language of the Union is Russian …
…
Bridge entity
Non-bridge entity
Target entity
Non-Target entity

Figure 7: An example of data preprocessing.

Text path	Ecrim	Ours w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$	Ours
Text path 1 [head document] **Merovingian script**, is named after an abbey in **Western France**, the **Luxeuil Abbey**, founded by the Irish missionary **St Columba** ca.590 ... [tail document]: ...The **Catholic Encyclopedia** (1913) concludes that the **Salome** of **Mark** 15: 40 is probably identical with the mother of the sons of **Zebedee** in **Matthew**; the latter is also mentioned in **Matthew** 20:20 ...	NA ✗	described by ✓	described by ✓
Text path 2 [head document]: ... **Battambang** is the capital city of **Battambang** province founded in the 11th century, **Lao Thai**, and **Chinese** ... The city is situated on the **Sangkae River** ... [tail document]: ... Textile and garment factories were built by **Chinese** investors, and the railway line was extended to **Poipet** ... Here are some facts and trivia. On the map of the world, **Asia** terminated in its southeastern point in a cape ...	NA ✗	NA ✗	continent ✓

Table 9: Two text paths with predicted results sampled from the test set of CodRED dataset. We use the same style to mark the text paths, where the **target entities**, **bridge entities**, and **non-bridge entities** are marked in green, blue, and orange respectively. In text path 1, with the help of non-bridge entities, both our model and ours w/o $\overline{y}_{rela}$, $\overline{y}_{bias}$ can predict the correct relation. In text path 2, we can find the proposed debiasing strategy helps our model for relation prediction.