A Novel Table-to-Graph Generation Approach for Document-Level Joint Entity and Relation Extraction

Ruoyu Zhang[1], **Yanzeng Li**[1], **Lei Zou**[1,2*]
[1]Wangxuan Institute of Computer Technology, Peking University. Beijing, China
[2]TopGraph.AI
{ry_zhang, zoulei}@pku.edu.cn
liyanzeng@stu.pku.edu.cn

Abstract

Document-level relation extraction (DocRE) aims to extract relations among entities within a document, which is crucial for applications like knowledge graph construction. Existing methods usually assume that entities and their mentions are identified beforehand, which falls short of real-world applications. To overcome this limitation, we propose TAG, a novel table-to-graph generation model for joint extraction of entities and relations at document-level. To enhance the learning of task dependencies, TAG induces a latent graph among mentions, with different types of edges indicating different task information, which is further broadcast with a relational graph convolutional network. To alleviate the error propagation problem, we adapt the hierarchical agglomerative clustering algorithm to back-propagate task information at decoding stage. Experiments on the benchmark dataset, DocRED, demonstrate that TAG surpasses previous methods by a large margin and achieves state-of-the-art results[1].

1 Introduction

Relation extraction (RE) is the task to extract relational facts from natural language text, which plays a crucial role in various downstream tasks, e.g. knowledge graph construction and question answering (Yih et al., 2015; Trisedya et al., 2019; Li and Zou, 2022). Early studies mostly focus on sentence-level RE, i.e. predicting relations among entities in one single sentence. However, in real-world scenarios such as Wikipedia articles or scientific papers, large amounts of relational facts are expressed across multiple sentences, which necessitate inter-sentence reasoning skills. Hence, recent efforts have been moving towards the more realistic document-level RE (DocRE) (Yao et al., 2019; Nan et al., 2020; Zhou et al., 2021).

Juan Balboa Boneke (9 June 1938 – 10 March 2014) was an Equatorial Guinean politician and writer. … After his exile, he settled down in Valencia with his second wife and her family. Balboa Boneke died from renal problems, coupled with a three-year depression caused by the death of his wife, on 10 March 2014 in Valencia, Spain.

Subject: *Balboa Boneke*	**Object:** *Equatorial Guinean*
Relation: country of citizenship	
Subject: *Balboa Boneke*	**Object:** *Valencia*
Relation: place of death	
Subject: *Valencia*	**Object:** *Spain*
Relation: country	

Figure 1: An example adapted from the DocRED dataset. Mentions refer to the same entity are in same color. We omit some relations and denote some entities with underline for clarity.

Despite the rapid progress, most previous DocRE methods solely focus on the task of relation extraction, which assumes that entities and their corresponding mentions are given beforehand. As shown by Figure 1, to extract both of entities and relations at document-level, a natural idea is to use a pipeline approach. Traditionally, it first divide the whole task into subtasks of mention extraction (ME), coreference resolution (COREF) and relation extraction (RE), then use separate models to conduct each task step by step (Zaporojets et al., 2021). However, the pipeline framework ignores the underlying dependencies among subtasks, which may lead to suboptimal performance. Some progress on jointly considering the subtasks has been made (Eberts and Ulges, 2021; Xu and Choi, 2022), yet, previous attempts still model the tasks of COREF and RE separately, inducing possible bias at both encoding and decoding stages. On the one hand, these methods still suffer from the problem with lack of information sharing. They either

*Corresponding author.
[1]https://github.com/ridiculouz/TaG

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 10853–10865

completely rely on the shared language model (e.g. BERT) at representation level (Eberts and Ulges, 2021) , or only consider one-way information flow from RE to COREF and neglect other cross-task dependencies (Xu and Choi, 2022). On the other hand, prior approaches mostly employ the pipeline-style decoding, which first recognize mention spans and form entity clusters, then perform relation classification for each entity pair. Such routine is not only time consuming, but faces with the error propagation problem (Li and Ji, 2014). The results of entity extraction may affect the performance of relation extraction and lead to cascading errors. Xu and Choi (2022) attempt to use a regularization term in COREF scorer to mitigate this issue, but the problem is still not fully resolved.

In this work, we propose TAG, a novel table-to-graph generation model, to address these aforementioned challenges. We first unify both tasks of COREF and RE with the classic table filling framework (Miwa and Sasaki, 2014; Gupta et al., 2016). We then devise a following table filler to encode original texts and make predictions for both tasks at a coarse level. Regarding mentions as nodes, we dynamically build two corresponding coreference and relation graphs, where the edges are weighted by the confidence scores of table filler. Besides, to alleviate the long-term dependency problem as well as explicitly model the syntactic information, we construct a syntactic graph over mentions. Given these three subgraphs, TAG regards them as three different types of edges and uses a relational graph convolutional network (R-GCN, Schlichtkrull et al., 2018) to model implicit task dependencies at a fine level. Unlike previous multi-task systems that solely share span representations directly from the language model, our coarse-to-fine framework leverages rich node representations by propagating information through semantic and syntactic links.

Intuitively, mentions within the same entity cluster should establish similar relation links with other entities (Xu and Choi, 2022). To avoid the error propagation problem, we exploit this postulation and adapt the hierarchical agglomerative clustering (HAC) algorithm to cluster mentions. The core of HAC is the computation of coreference distance between each cluster pair. To back-propagate relational information, we compute the relation vectors of nodes and use the average Hamming distances among different clusters as additional penalty.

We evaluate TAG on DocRED (Yao et al., 2019), a widely-adopted DocRE benchmark. Experiments show that: (1) The coarse-grained table filler baseline establishes competitive results, as compared with previous methods. (2) The fine-grained information propagation module and enhanced HAC decoding algorithm can effectively promote cross-task interactions and better alleviate the error propagation problem. (3) Our proposed TAG achieves new state-of-the-art and outperforms prior approaches by a large margin. We also report the first result of joint entity and relation extraction on Re-DocRED (Tan et al., 2022), a revised version of DocRED, for future research.

Our contributions can be summarized as follow:

- We unify the tasks of COREF and RE in document-level joint entity and relation extraction with a table filling framework, and propose a novel table-to-graph generation method TAG to facilitate information sharing. During the decoding stage, we adapt the HAC algorithm to enhance COREF with RE predictions, thereby mitigating the issue of error propagation.
- We demonstrate that TAG surpasses previous methods and achieves new state-of-the-art results on the standard DocRE benchmark.

2 Problem Formulation

Given a document D comprised of L tokens, our goal is to jointly extract all entities and relations in an end-to-end manner. As an entity may occur multiple times in the document with different mentions, the joint extraction process can be naturally divided into three subtasks:

- Mention extraction (ME), which extracts all possible spans $\mathcal{M} = \{m_i\}_{i=1}^{M}$ for entities from original document, where a span is defined as a continuous sequence of words;
- Coreference resolution (COREF), which groups the local mentions into entity clusters $\mathcal{E} = \{e_i\}_{i=1}^{E}$, where $e_i = \{m_j^i\}_{j=1}^{N_{e_i}}$;
- Relation extraction (RE), which predicts a subset from a pre-defined relation set $\mathcal{R} \cup \{\perp\}$ ($\perp$ denotes no relation) between the entity pairs $(e_h, e_t)_{h,t=1,\ldots,E;h\neq t}$.

Unlike prior works, we formulate the tasks of COREF and RE with the table filling framework,

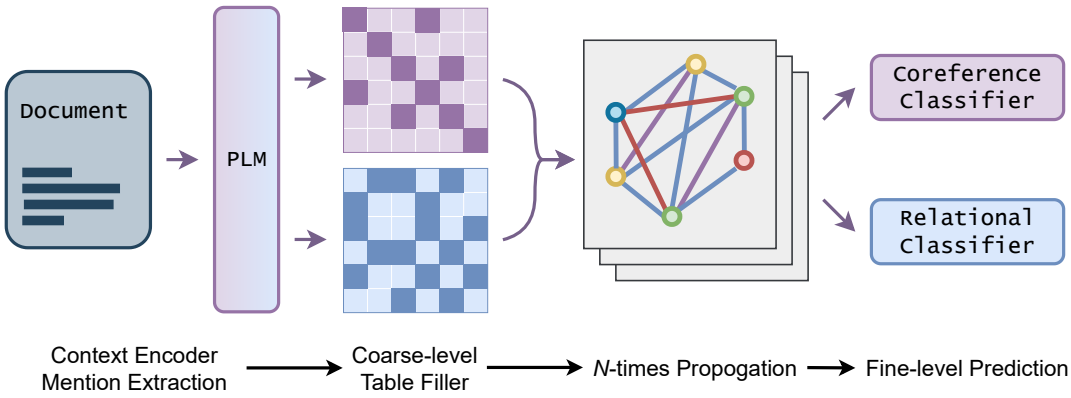

Figure 2: Overall architechture of TaG. Given a document, it first conducts mention extraction separately, and then use a table filler to predict coreference scores (purple matrix) and relational scores (blue matrix) at a coarse level. A mention graph with coreference, relational and syntactic edges is then built, based on which we leverage R-GCN to propagate information. We predict the final results with the fine-level mention representations.

i.e. multi-class classification between each mention pair (m_i, m_j). We maintain a table $T^{|M|\times|M|}$ to represent mention pairs and employ a shared representation for both tasks.

We assign COREF label $y_c^{(i,j)} \in \{0, 1\}$ and RE label $y_r^{(i,j)} \subseteq \mathcal{R} \cup \{\perp\}$ for each cell in the table, respectively. For COREF, we use 1/0 to denote whether a mention pair belongs to the same entity. For RE, we transfer the entity-level label to mention-level, where mention pair (m_i, m_j) is tagged with the same relations of their belonging entities (e_h, e_t), with $m_i \in e_h, m_j \in e_t$.

3 Methodology

Figure 2 shows the overall architecture of TaG. TaG first conducts ME to predict mention spans (§ 3.1), after which, it jointly learns the tasks of COREF and RE with a table-to-graph generation model (§ 3.2). We will also detail the multi-task training process in § 3.3 and enhanced decoding algorithm in § 3.4.

3.1 Mention Extractor

We cast the problem of entity mention extraction as a sequence tagging task with `BIO` label. Though span-based methods are more prevalent due to their stronger expressive power, they usually demand $\mathcal{O}(L^2)$ time complexity, while sequence-based methods only take linear time. Since the task of DocRE contains few overlapped mentions[2], we adopt the sequential method for efficiency.

Following Devlin et al. (2019), we leverage pre-trained language model (PLM) to convert the tokens in document into vectorized features, and use a classifier to predict the `BIO` label for each token. We denote the extracted mentions by $\{m_i\}_{i=1}^{M}$.

3.2 Table-to-Graph Generation

3.2.1 Biaffine Table Filler

Given a document $D = [w_i]_{i=1}^{L}$ and mentions $\{m_i\}_{i=1}^{M}$, we build the table representation of each mention pair. We adopt the entity marker strategy (Baldini Soares et al., 2019), which inserts a special token "*" at the start and end of each mention. We then use a separate PLM[3] to obtain the contextual representations $\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_L]^\top, \mathbf{h}_i \in \mathbb{R}^d$ and the multi-head attention $\mathbf{A} \in \mathbb{R}^{H\times L\times L}$:

$$\mathbf{H}, \mathbf{A} = \texttt{PLM}([w_1, \ldots, w_L]),$$

where $\mathbf{A}$ is the multi-head attention matrix in the last transformer layer. We take the embedding of start token "*" as mention embedding. To capture related context for mention pair (m_i, m_j), we apply the localized context pooling technique to compute context embedding $\mathbf{c}^{(i,j)}$ (Zhou et al., 2021):

$$\mathbf{q}^{(i,j)} = \sum_{k=1}^{H} \mathbf{A}_k^i \circ \mathbf{A}_k^j,$$

$$\mathbf{c}^{(i,j)} = \mathbf{H}^\top \frac{\mathbf{q}^{(i,j)}}{\mathbf{1}^\top \mathbf{q}^{(i,j)}},$$

[2] In the standard benchmark DocRED, only 0.2% mentions are overlapped, and this phenomenon is usually caused by annotation errors as well.

[3] Our preliminary experiments show that multi-tasking ME brings marginal benefits. So we conduct ME as an independent task and use separate PLM in ME and COREF/RE.

where $\circ$ refers to the Hadamard product and $\mathbf{A}_k^i, \mathbf{A}_k^j \in \mathbb{R}^L$ are the attention weights of m_i, m_j in the k^{th} attention head, respectively. $\mathbf{c}^{(i,j)}$ is aggregated from tokens with high attention towards both m_i and m_j, and hence is likely to be important to both of them.

Let $\mathbf{h}_i, \mathbf{h}_j$ be the hidden features of m_i, m_j from PLM. We first project $\mathbf{h}_i, \mathbf{h}_j$ and $\mathbf{c}^{(i,j)}$ into head and tail features:

$$\mathbf{z}_i^{(i,j)} = \texttt{tanh}(\mathbf{W}_h\mathbf{h}_i + \mathbf{W}_{ch}\mathbf{c}^{(i,j)}),$$
$$\mathbf{z}_j^{(i,j)} = \texttt{tanh}(\mathbf{W}_t\mathbf{h}_j + \mathbf{W}_{ct}\mathbf{c}^{(i,j)}),$$

where $\mathbf{W}_h, \mathbf{W}_{ch}, \mathbf{W}_t, \mathbf{W}_{ct} \in \mathbb{R}^{d\times d}$ are trainable parameters. We then employ a biaffine attention model (Dozat and Manning, 2017; Wang et al., 2021) to convert mention features into a table $\mathbf{S} \in \mathbb{R}^{M\times M}$ of scalar scores denoting either coreference or relational links:

$$s^{(i,j)} = \mathbf{z}_i^{(i,j)}\mathbf{W}_1\mathbf{z}_j^{(i,j)} + \mathbf{w}_2^\top(\mathbf{z}_i^{(i,j)} \oplus \mathbf{z}_j^{(i,j)}) + b,$$

where $\mathbf{W}_1 \in \mathbb{R}^{d\times d}, \mathbf{w}_2 \in \mathbb{R}^{2d}, b \in \mathbb{R}$ are trainable parameters, $\oplus$ denotes concatenation. We predict coreference and relational scores $\mathbf{S}_{tc}, \mathbf{S}_{tr}$ respectively with shared representations $\mathbf{z}$. Specifically, $s_{tr}^{(i,j)}$ is labeled with 1 if the RE label $y_r^{(i,j)} \neq \{\perp\}$ otherwise 0.

3.2.2 Latent Graph Construction

Coreference and Relational Graphs. After obtaining the coreference and relational scores $\mathbf{S}_{tc}, \mathbf{S}_{tr}$, we normalize each table with respect to column:

$$\mathbf{G}_c = \texttt{Softmax}(\mathbf{S}_{tc}),$$
$$\mathbf{G}_r = \texttt{Softmax}(\mathbf{S}_{tr}).$$

We take $\mathbf{G}_c$ and $\mathbf{G}_r$ as the dynamic weighted graphs of coreference and relational links predicted by our previous modules. Each cell $g^{(i,j)}$ represents the weight of directed edge $m_i \rightarrow m_j$.

Syntactic Graph. To enhance learning of structured knowledge underlying natural language, we seek to explicitly introduce syntactic information into mention graph. Ideally, syntactic links can effectively encode local contexts, which can be further broadcast via coreference or relational links. Thus, it enables the model to learn long-term dependencies at a fine level.

There are several optional ways to build the desired syntactic graph. For instance, an intuitive solution is to transfer the dependency tree over words to a graph, with mentions being the nodes. Since dependency tree only reveals intra-sentence clues, previous works (Christopoulou et al., 2019; Zeng et al., 2020) usually leverage co-occurrence information instead. Following this practice, our syntactic graph $\mathbf{G}_s$ connects all mentions within the same sentence using bidirectional edges.

3.2.3 Propagating Information with R-GCN

To consider the interactions between the tasks of COREF and RE, and to incorporate explicit syntax information, we propose an information propagation module to refine mention representations.

Specifically, we regard the latent graphs $\mathbf{G}_c, \mathbf{G}_r$ and $\mathbf{G}_s$ as three different types of edges over the mention graph. We then apply a relational graph convolutional network on the mention graph to aggregate neighbor features along different types of edges. Given node $\mathbf{x}_i$ at the l^{th} layer, the update process is calculated by

$$\mathbf{x}_i^{(l+1)} = \texttt{tanh}\Big(\sum_{t\in\{c,r,s\}}\sum_{j=1}^{M} g_t^{(i,j)}\mathbf{W}_t^l\mathbf{x}_j^l + \mathbf{b}_t^l\Big),$$

where t is the type of edge, $g_t^{(i,j)}$ represents the weight of directed edge $m_i \rightarrow m_j$, and $\mathbf{W}_t^l, \mathbf{b}_t^l$ are trainable parameters. We initialize node embedding $\mathbf{x}_i^0$ as the hidden feature $\mathbf{h}_i$ of mention m_i.

In contrast to previous Joint IE[4] approaches, which either propagate task information in a pipeline manner (DyGIE, Luan et al., 2019), or only consider one-way information flow (Xu and Choi, 2022), our module integrates cross-task information in parallel and extracts relevant mention features for both tasks.

3.2.4 Classifier

After N times of propagation, we use the refined mention embeddings $\mathbf{x}_i^N, \mathbf{x}_j^N$ and context embedding $\mathbf{c}^{(i,j)}$ to predict the COREF score $s_{gc}^{(i,j)}$ and RE score $s_{gr}^{(i,j)}$:

$$\mathbf{v}_i^{(i,j)} = \texttt{tanh}(\mathbf{U}_h\mathbf{x}_i^N + \mathbf{U}_{ch}\mathbf{c}^{(i,j)}),$$
$$\mathbf{v}_j^{(i,j)} = \texttt{tanh}(\mathbf{U}_t\mathbf{x}_j^N + \mathbf{U}_{ct}\mathbf{c}^{(i,j)}),$$
$$s_{gc}^{(i,j)} = \texttt{CorefBiaff}(\mathbf{v}_i^{(i,j)}, \mathbf{v}_j^{(i,j)}),$$
$$s_{gr}^{(i,j)} = \texttt{ReBiaff}(\mathbf{v}_i^{(i,j)}, \mathbf{v}_j^{(i,j)}),$$

[4]Joint information extraction.

where $\mathbf{U}_h, \mathbf{U}_{ch}, \mathbf{U}_t, \mathbf{U}_{ct} \in \mathbb{R}^{d\times d}$ are trainable parameters, and the n-dimensional biaffine function is defined as

$$\texttt{Biaff}(\mathbf{x}, \mathbf{y}) \coloneqq \mathbf{x}\mathbf{U}_1^\top \mathbf{y} + \mathbf{U}_2(\mathbf{x} \oplus \mathbf{y}) + \mathbf{b},$$

where $\mathbf{U}_1 \in \mathbb{R}^{n\times d\times d}, \mathbf{U}_2 \in \mathbb{R}^{n\times 2d}, \mathbf{b} \in \mathbb{R}^n$ are trainable parameters. Note that $n = 1$ for the task of COREF and $n = |\mathcal{R}| + 1$ for RE, where we use a dummy class TH to learn a dynamic threshold for multi-label classification (Zhou et al., 2021). At test time, relation types with scores higher than the TH class are predicted as output $\hat{y}_r^{(i,j)}$. In cases where no such class exists, the classifier returns $\{\perp\}$.

3.3 Training

We perform multi-task training and optimize the joint loss for all components. We detail the training objectives and label construction for each module as follows.

Table Encoder. Given mention pair (m_i, m_j), the table encoder predicts coreference and relational links in the form of scalar scores $s_{tc}^{(i,j)}, s_{tr}^{(i,j)}$. For coreference links, we directly use COREF label $y_c^{(i,j)}$ as gold label. For relational links, we define $y_{rbinary}^{(i,j)} \coloneqq \mathbb{1}(y_r^{(i,j)} \neq \{\perp\})$[5], denoting whether any relation (e_h, r, e_t) exists, with $m_i \in e_h, m_j \in e_t$. We convert $\mathbf{S}_c, \mathbf{S}_r$ to probability with the sigmoid function σ and optimize with binary cross-entropy loss $\mathcal{L}_{tc}, \mathcal{L}_{tr}$.

Coreference Resolution. The training objective and label for fine-level coreference resolution are identical to those for coreference link prediction in table encoder. The sole difference is that it takes the refined mention representations as input. We denote the loss as $\mathcal{L}_{gc}$.

Relation Extraction. For (m_i, m_j), we divide the relation set $\mathcal{R}$ into two splits: positive set $\mathcal{P}$ that contains relation x exists between (m_i, m_j), and negative set $\mathcal{N} = \mathcal{R} - \mathcal{P}$. We apply the adaptive-thresholding loss (Zhou et al., 2021) to learn the RE classifier:

$$l^{(i,j)} = -\sum_{x\in\mathcal{P}} \log\left(\frac{\exp(s_x^{(i,j)})}{\sum_{x'\in\mathcal{P}\cup\{\text{TH}\}}\exp(s_{x'}^{(i,j)})}\right) - \log\left(\frac{\exp(s_{\text{TH}}^{(i,j)})}{\sum_{x'\in\mathcal{N}\cup\{\text{TH}\}}\exp(s_{x'}^{(i,j)})}\right),$$

[5]Indicator function.

Algorithm 1: HAC Decoding Algorithm

Input: Mention set $\mathcal{M}$, threshold t
Output: A set of entity clusters $\mathcal{C}$

```
// Initialization
for m_i ∈ M do
    C_i ← {m_i}
// Recursively merge clusters
repeat
    for C_x, C_y ∈ C, C_x ≠ C_y do
        D^(x,y) ← D_c^(x,y) + ρ · D_r^(x,y)
    (C_x, C_y) ← arg min_(C_x,C_y) D^(x,y)
    D_min ← D^(x,y)
    if D_min ⩽ t then
        Merge C_x and C_y
until D_min > t
```

and we sum over all mention pairs to calculate fine-level relation extraction loss $\mathcal{L}_{gr}$.

Finally, we jointly optimize TAG with

$$\mathcal{L} = \mathcal{L}_{tc} + \mathcal{L}_{tr} + \alpha \cdot (\mathcal{L}_{gc} + \mathcal{L}_{gr}),$$

where α is a hyperparameter balancing coarse-level and fine-level loss.

3.4 Decoding

To avoid the error propagation problem inherent in pipeline decoding, we aim to design a decoding algorithm such that upstream task (COREF) can efficiently utilize downstream task information (RE).

Entity Cluster Decoding. We decode entity clusters based on the hierarchical agglomerative clustering (HAC) algorithm, as described in Algorithm 1. The core of HAC is to measure the distance D between two clusters C_x and C_y. We break down D into two parts: coreference distance D_c and relational distance D_r. We use the average linkage to compute D_c as

$$D_c = \frac{1}{|C_x|\cdot|C_y|}\sum_{m_i\in C_x}\sum_{m_j\in C_y}(1-\sigma(s_{gc}^{(i,j)})).$$

At training stage, ground-truth relation $y_r^{(i,k)}$ and $y_r^{(j,k)}$ are identical if m_i and m_j belong to the same entity, for all $m_k \in \mathcal{M}$. Therefore, for a well-trained model, mentions within the same entity cluster should establish similar relation links with other entities. We exploit this clue as the connection between COREF and RE. Let the predicted RE

Method	Encoder	ME	COREF	RE	
				F1	Ign F1
KB-IE (Verlinden et al., 2021)	LSTM	-	83.6	25.7	-
JEREX (Eberts and Ulges, 2021)	BERT-base	92.99*	82.79*	40.38*	-
seq2rel (Giorgi et al., 2022)	BERT-base	-	-	38.2*	-
Pipeline (Xu and Choi, 2022)	SpanBERT-base	92.56	84.09	38.29	35.88
Joint (Xu and Choi, 2022)	SpanBERT-base	93.34	84.79	38.94	36.64
JointM+GPGC (Xu and Choi, 2022)	SpanBERT-base	93.35	84.96	40.62	38.28
TABLEFILLER	BERT-base	93.56 / 92.89	84.77 / 84.34	40.92 / 39.10	39.09 / 37.30
	RoBERTa-base	**93.63 / 92.95**	85.87 / 85.49	42.00 / 40.92	40.09 / 38.97
TAG	BERT-base	93.56 / 92.89	85.07 / 84.75	41.87 / 40.65	39.82 / 38.27
	RoBERTa-base	**93.63 / 92.95**	**86.03 / 85.67**	**43.16 / 42.28**	**41.13 / 40.28**
TAG	RoBERTa-large	**93.84 / 93.32**	**86.37 / 85.87**	**44.97 / 43.21**	**42.88 / 41.22**

Table 1: Overall performance on DocRED. Previous methods only report results on test set, while we report results on both test/dev set, respectively. In particular, JEREX and seq2rel use a custom split of DocRED, so their results are not directly comparable and only serve for reference.

label $\hat{y}_r^{(i,j)}$ be a $|\mathcal{R}|$-dimensional 0-1 vector, where each digit indicates the presence of one relation type. We define the relation vector $\mathbf{r}_i \in \mathbb{R}^{2M \times |\mathcal{R}|}$ as

$$\mathbf{r}_i = [\hat{y}_r^{(i,1)}, \ldots, \hat{y}_r^{(i,M)}, \hat{y}_r^{(1,i)}, \ldots, \hat{y}_r^{(M,i)}]^\top.$$

We use the average Hamming distance between each mention pair in cluster C_x, C_y as D_r:

$$D_r = \frac{1}{|C_x||C_y|} \sum_{m_i \in C_x} \sum_{m_j \in C_y} \sigma(\texttt{Hamming}(\mathbf{r}_i, \mathbf{r}_j)).$$

Relation Triple Decoding. Given two entities e_1 and e_2, we predict their relation label with the majority voting mechanism. For relation x, the final prediction is determined by

$$\hat{y}_x^{(e_1,e_2)} = \mathbb{1}((\sum_{m_i \in e_1} \sum_{m_j \in e_2} \hat{y}_x^{(i,j)}) > \frac{|e_1| \cdot |e_2|}{2}).$$

4 Experiments

4.1 Setup

Dataset. We evaluate TAG on **DocRED** (Yao et al., 2019) and **Re-DocRED** (Tan et al., 2022). DocRED is a large-scale human-annotated dataset for DocRE constructed from Wikipedia and Wikidata. It covers a wide range of documents from general domain, with 3,053 documents for training, 1,000 for development, and 1,000 for test, respectively. DocRED contains 96 relation types, 132,375 entities and 63,427 relation instances. Since the original dataset is incomplete, i.e. there exists a considerable amount of false negative samples, Tan et al. (2022) provide a revised version Re-DocRED on training and validation set, with 120,664 relation instances. Notably, we report the first joint extraction result on Re-DocRED for future reference.

Metrics. Following prior works (Eberts and Ulges, 2021; Xu and Choi, 2022), we report the performance of all three subtasks for detailed analysis. Specifically, our results include (1) mention extraction (ME) in mention-level F1 score, (2) coreference resolution (COREF) in averaged F1 score of MUC, B^3, and $CEAF_{\phi_4}$, and (3) relation extraction (RE) in hard entity-level F1 and Ign F1 scores, where Ign F1 measures the F1 score excluding the relational facts shared by training and validation/test sets.

4.2 Overall Performance

Baselines. We compare TAG with various baselines for joint extraction. Early approaches take LSTM as context encoder. Built on top of it, Verlinden et al. (2021) introduce **KB-IE**, which integrates background information of knowledge base (Wikipedia and Wikidata) into a joint IE model. Recent methods usually finetune PLM to learn richer features. Xu and Choi (2022) implement the standard **pipeline** method, as well as a **joint** method with shared encoder and joint loss. They also propose **JointM+GPGC** to enable one-way information flow from RE to COREF. Eberts and Ulges (2021) present **JEREX**, which incorporate multi-instance learning to enhance RE performance. Giorgi et al. (2022) develop a sequence-

Method	ME	COREF	RE	
			F1	Ign F1
TABLEFILLER	**93.42**	86.27	48.35	47.30
TAG	**93.42**	**86.49**	**49.34**	**48.21**
TABLEFILLER	**92.91**	85.25	48.94	48.02
TAG	**92.91**	**85.61**	**49.38**	**48.47**

Table 2: Performance on Re-DocRED, which takes RoBERTa$_{base}$ as encoder. The former/latter two lines denote results on dev/test set, respectively.

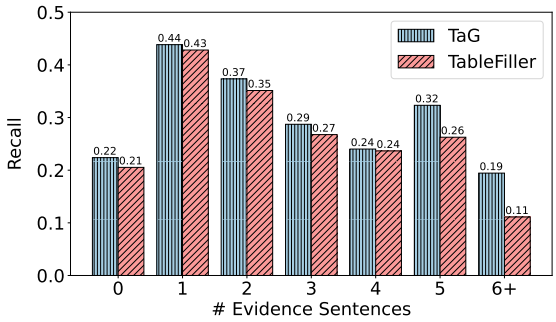

Figure 3: Recall of relations over different numbers of evidence sentences. We compare TAG and TABLEFILLER with RoBERTa$_{base}$ on DocRED dev set.

to-sequence model with copy mechanism, **seq2rel**, with inferior performance but higher efficiency. Besides, we also devise a strong baseline, **TableFiller**, which ablates the graph module and adopts simple heuristic decoding algorithm, i.e. it only comprises a mention extractor, a biaffine encoder, and a classifier.

Table 1 depicts the overall performance of TAG on DocRED, in comparison to other baselines. We can observe that TABLEFILLER-BERT$_{base}$ marginally outperforms previous methods and establishes a competitve basis, which demonstrates the efficacy of the table filling framework. TAG-BERT$_{base}$ further advances it by consistent improvements on all three subtasks. Following Xu and Choi (2022), we replace BERT$_{base}$ with a stronger variant, RoBERTa$_{base}$, of the same size. TAG-RoBERTa$_{base}$ attains substantial improvements of 1.07 in COREF F1 and 2.54/2.85 in RE F1/Ign F1 over SOTA on the test set. This suggests that TAG is better at capturing important information within the document-level context and across different subtasks. We also present TAG-RoBERTa$_{large}$ to explore the boundaries of joint extraction performance, which reaches 93.84 in ME F1, 86.37 in COREF F1 and 44.97/42.88 in RE F1/Ign F1 on the test set, respectively.

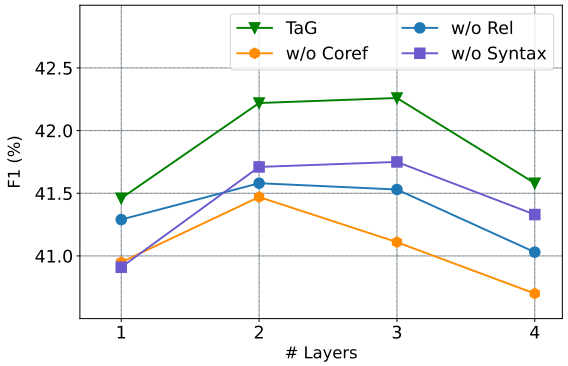

Figure 4: Relation extraction F1 of TAG variants with different numbers of graph layers on DocRED dev set.

Table 2 shows the performance of TABLEFILLER and TAG on Re-DocRED. In comparison to DocRED, the same methods yield similar performances in coreference resolution, but improve by a large margin in relation extraction, which aligns with previous findings (Tan et al., 2022). Regarding the difference in architectures, TAG consistently outperforms TABLEFILLER in all subtasks on both dev and test sets, highlighting the effectiveness of TAG for document-level joint extraction.

4.3 Analysis on Reasoning Skills

A major challenge for document-level RE is the requirement of rich reasoning skills, e.g. common-sense reasoning and logical reasoning (Yao et al., 2019). One indicator to distinguish the reasoning type is the amounts of evidence sentences. To understand the merits of TAG, we visualize the recall of relations over different amounts of evidence sentences, as shown by Figure 3.

Relation instance with 0 evidence can only be inferred from common-sense knowledge, either from PLM knowledge or training corpus. TAG outperforms TABLEFILLER on such type of instances by 1.8% with the same encoder, which demonstrates the stronger ability of **common-sense reasoning**. TAG also consistently surpasses TABLEFILLER on a vast amount of relations with 2-4 evidence sentences, which either needs to (1) distinguish coreferential mentions within multiple sentences, or (2) perform logical reasoning over bridge entities. This reveals that the graph module and decoding algorithm are beneficial for both **coreference reasoning** and **multi-hop logical reasoning**. Finally, TAG substantially improves the recall of relations that require much evidence (6.0% for 5 sentences and

ρ	0	0.05	0.1	0.2	0.3
Averaged F1	85.36	85.46	**85.67**	85.51	85.44
Hard F1	82.75	82.81	**83.06**	82.92	82.73

Table 3: F1 scores of TAG-RoBERTa$_{\text{base}}$ on DocRED dev set with different hyperparameter ρ.

	silver	score$_c$	score$_r$
silver	1.00	0.91	-0.72
score$_c$	-	1.00	-0.74
score$_r$	-	-	1.00

Table 4: Pearson's r for silver COREF label, corefenrece score and relational penalty on DocRED dev set. See Appendix B for details.

8.3% for more than 6 sentences), indicating that TAG is superior at **complex logical reasoning**.

4.4 The Impact of Graph Propagation

Figure 4 shows the effects of graph propagation on relation extraction F1 score, where `-Coref`, `-Rel` and `-Syntax` denote the removal of the corresponding type of edges, respectively. It can be seen that the F1 scores of all models usually peak at 2/3 graph layers, and then decrease drastically. We hypothesize that a greater depth of layers facilitates the dissemination of information to a broader range, whereas the gradient vanishing problem counteracts this advantage (Li et al., 2019). Besides, all ablation models perform worse than TAG with full channels, indicating that all types of edges contribute to better reasoning.

As the depth of layers and types of edges influence RE F1 dramatically, in contrast, these different settings do not pose much impact on coreference resolution. We will dive deeper into this question in the following subsection.

4.5 Effectiveness of Decoding

To verify the effectiveness of our entity cluster decoding algorithm, we compare the performance of coreference resolution with different balancing hyperparameter ρ in Table 3. Apart from the averaged F1 score of MUC, B^3, and CEAF$_{\phi_4}$, we also report the hard entity-level F1 score for transparently demonstrating the entity extraction performance. It can be seen that $\rho = 0.1$ yields the optimal performance with a 0.3% F1 gain in both metrics.

Despite that the performance of HAC decoding algorithm is boosted by the relational distance D_r, the observed improvement is not as substantial as anticipated. Besides, adjusting ρ does not influence much as well. These findings indicate that coreference resolution seems to be more robust with various settings. To understand such phenomenon, we conduct a correlation analysis among the silver COREF label and predicted scores, as shown by Table 4. While there exists a significant correlation of -0.72 between the relational penalty and the silver label, it is still well below the correlation between coreference score and silver label. This strong association partially accounts for the aforementioned results. It further shows that D_r can only serve as a modest refining signal for coreference resolution, and increasing ρ above the threshold may hurt COREF performance.

5 Related Works

Document-level extraction and joint extraction are two important topics in the field of IE. Our work lies at the intersection of these two lines, which aims to jointly extract entities and relations, two core elements of IE, at document-level.

Document-level RE. Current methods in DocRE can be mainly divided into two categories: (1) Graph-based methods, which first construct a document graph of heterogeneous nodes (e.g. mention, entity, sentence) with heuristic rules, and then use GNN to perform inference on the graph (Christopoulou et al., 2019; Nan et al., 2020; Zeng et al., 2020). (2) Transformer-based methods, which exploits pretrained language model to learn cross-sentence relations either implicitly or explicitly. Various techniques have been proposed, e.g. adaptive threshold (Zhou et al., 2021) and evidence retrieval (Huang et al., 2021; Xie et al., 2022). Recently, pioneers have attempted to develop end-to-end models that extracts entities and relations jointly at document-level, which is more practical and brings more challenges (Eberts and Ulges, 2021; Xu and Choi, 2022; Giorgi et al., 2022).

Joint information extraction. Early studies usually model Joint IE in a pipeline manner (Chan and Roth, 2011; Luan et al., 2019), which ignores the underlying correlation within different tasks, suffering from cascading errors and exposure bias. To address these problems, in one direction, some recent researches seek to integrate multiple subtasks by sharing information and building up implicit cross-task interaction (Zhang et al., 2020; Yan et al., 2021). In another direction, table filling strategy

has been developed, as it casts subtasks (usually NER and RE) as unified table to fill with, which explicitly leverages the interactions among subtasks (Miwa and Sasaki, 2014; Gupta et al., 2016; Wang et al., 2021).

6 Conclusion

In this paper, we propose TAG, a novel table-to-graph generation model, to jointly extract entities and relations within a document. Different from prior approaches, we unify the tasks of coreference resolution and relation extraction with a table filling framework, and leverage a coarse-to-fine strategy to facilitate information sharing among these subtasks. To avoid the error propagation problem, we adapt the HAC algorithm to enhance COREF with RE predictions at decoding stage. Experimental results on the widely-adopted benchmark, DocRED, demonstrate that TAG significantly outperforms previous methods. Further analysis also confirms the effectiveness of the modules in our model.

Limitations

One major limitation of our work is that our experiments are only conducted on DocRED and Re-DocRED that consist of documents from general domain. Yet, information extraction has many broader applications in specific domains, e.g. biomedical data. We plan to adapt TAG to some biomedical datasets, like CDR (Li et al., 2016) and GDA (Wu et al., 2019), in the future.

Besides, since TAG consists of a number of modules and use PLM as encoder, the training process takes relatively more time and computational resources than dedicated DocRE model that only extract relations. We concern that it may affect the scalability with larger amount of either data or parameters.

Ethics Statement

We use DocRED and Re-DocRED in our experiments, and we adhere to their user agreements and licenses. These datasets are constructed from Wikipedia, which we expect to have few offensive contents or leaked privacy information.

We shall point out that our system may generate false results due to the nature of neural networks, and may be biased in the cases of domain shift or out-of-distribution. We concern that appropriate quality control is needed in downstream applications, like knowledge base construction.

Acknowledgements

We would like to appreciate the reviewers for their valuable comments that help us to improve this manuscript. This work was supported by NSFC under grant 61932001 and U20A20174. Lei Zou is the corresponding author of this paper.

References

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, Florence, Italy. Association for Computational Linguistics.

Yee Seng Chan and Dan Roth. 2011. Exploiting syntactico-semantic structures for relation extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 551–560, Portland, Oregon, USA. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. 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 4925–4936, Hong Kong, China. Association for Computational Linguistics.

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

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *International Conference on Learning Representations*.

Markus Eberts and Adrian Ulges. 2021. An end-to-end model for entity-level relation extraction using multi-instance learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3650–3660, Online. Association for Computational Linguistics.

John Giorgi, Gary Bader, and Bo Wang. 2022. A sequence-to-sequence approach for document-level

relation extraction. In *Proceedings of the 21st Workshop on Biomedical Language Processing*, pages 10–25, Dublin, Ireland. Association for Computational Linguistics.

Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2537–2547, Osaka, Japan. The COLING 2016 Organizing Committee.

Kevin Huang, Peng Qi, Guangtao Wang, Tengyu Ma, and Jing Huang. 2021. Entity and evidence guided document-level relation extraction. In *Proceedings of the 6th Workshop on Representation Learning for NLP (RepL4NLP-2021)*, pages 307–315, Online. Association for Computational Linguistics.

Guohao Li, Matthias Muller, Ali Thabet, and Bernard Ghanem. 2019. Deepgcns: Can gcns go as deep as cnns? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.

Jiao Li, Yueping Sun, Robin J. Johnson, Daniela Sciaky, Chih-Hsuan Wei, Robert Leaman, Allan Peter Davis, Carolyn J. Mattingly, Thomas C. Wiegers, and Zhiyong Lu. 2016. BioCreative V CDR task corpus: a resource for chemical disease relation extraction. *Database*, 2016. Baw068.

Qi Li and Heng Ji. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, Baltimore, Maryland. Association for Computational Linguistics.

Yanzeng Li and Lei Zou. 2022. gbuilder: A scalable knowledge graph construction system for unstructured corpus.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3036–3046, Minneapolis, Minnesota. Association for Computational Linguistics.

Makoto Miwa and Yutaka Sasaki. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, Doha, Qatar. Association for Computational Linguistics.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1546–1557, Online. Association for Computational Linguistics.

Michael Schlichtkrull, Thomas N. Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *The Semantic Web*, pages 593–607, Cham. Springer International Publishing.

Qingyu Tan, Lu Xu, Lidong Bing, Hwee Tou Ng, and Sharifah Mahani Aljunied. 2022. Revisiting docred – addressing the false negative problem in relation extraction.

Bayu Distiawan Trisedya, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. 2019. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, Florence, Italy. Association for Computational Linguistics.

Severine Verlinden, Klim Zaporojets, Johannes Deleu, Thomas Demeester, and Chris Develder. 2021. Injecting knowledge base information into end-to-end joint entity and relation extraction and coreference resolution. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1952–1957, Online. Association for Computational Linguistics.

Yijun Wang, Changzhi Sun, Yuanbin Wu, Hao Zhou, Lei Li, and Junchi Yan. 2021. UniRE: A unified label space for entity relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 220–231, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2019. Huggingface's transformers: State-of-the-art natural language processing.

Ye Wu, Ruibang Luo, Henry C. M. Leung, Hing-Fung Ting, and Tak-Wah Lam. 2019. Renet: A deep learning approach for extracting gene-disease associations from literature. In *Research in Computational Molecular Biology*, pages 272–284, Cham. Springer International Publishing.

Yiqing Xie, Jiaming Shen, Sha Li, Yuning Mao, and Jiawei Han. 2022. Eider: Empowering document-level relation extraction with efficient evidence extraction and inference-stage fusion. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 257–268, Dublin, Ireland. Association for Computational Linguistics.

Liyan Xu and Jinho Choi. 2022. Modeling task interactions in document-level joint entity and relation extraction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5409–5416, Seattle, United States. Association for Computational Linguistics.

Zhiheng Yan, Chong Zhang, Jinlan Fu, Qi Zhang, and Zhongyu Wei. 2021. A partition filter network for joint entity and relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 185–197, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yuan Yao, Deming Ye, Peng Li, Xu Han, Yankai Lin, Zhenghao Liu, Zhiyuan Liu, Lixin Huang, Jie Zhou, and Maosong Sun. 2019. DocRED: A large-scale document-level relation extraction dataset. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 764–777, Florence, Italy. Association for Computational Linguistics.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He, and Jianfeng Gao. 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Klim Zaporojets, Johannes Deleu, Chris Develder, and Thomas Demeester. 2021. Dwie: An entity-centric dataset for multi-task document-level information extraction. *Information Processing & Management*, 58(4):102563.

Shuang Zeng, Runxin Xu, Baobao Chang, and Lei Li. 2020. Double graph based reasoning for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1630–1640, Online. Association for Computational Linguistics.

Ranran Haoran Zhang, Qianying Liu, Aysa Xuemo Fan, Heng Ji, Daojian Zeng, Fei Cheng, Daisuke Kawahara, and Sadao Kurohashi. 2020. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 236–246, Online. Association for Computational Linguistics.

Wenxuan Zhou, Kevin Huang, Tengyu Ma, and Jing Huang. 2021. Document-level relation extraction with adaptive thresholding and localized context pooling. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(16):14612–14620.

A Implementation

Our model is implemented based on PyTorch and HuggingFace's Transformer (Wolf et al., 2019). We leverage BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019) of different sizes as PLM encoder, and stack 2/3 layers of R-GCN for graph propagation for different settings/datasets. The hyperparameters α and ρ for training and decoding are set to 1 and 0.1, respectively. We optimize our model using AdamW (Loshchilov and Hutter, 2019) with learning rate 3e-5 for PLM and 1e-4 for other parameters, under a linear warmup for the first 4% steps. We train our model with a batch size of 4 for 50 epochs, which takes ~5 hours on a single A40 GPU. We use early stopping strategy for efficiency.

All experiments are conducted under 3 random seeds, and we report: (1) the result of model with best dev score for DocRED test set, since the evaluation is organized as a Codalab competition[6], (2) the average result of all three runs for DocRED dev set and Re-DocRED.

B Details for Correlation Analysis

We conduct the correlation analysis on dev set of DocRED with TAG-RoBERTa$_{\text{base}}$. The variables are constructed as follow:

- **Silver**. Given predicted mention spans, we assign silver label 1 for mentions that occur within the same gold entity, and 0 otherwise.
- **Score**$_c$. The probability of coreference link $\sigma(s_{gc})$.
- **Score**$_r$. $|\mathbf{s}_r^i - \mathbf{s}_r^j|_1$, which serves as a pairwise estimation of the Hamming distance. Particularly, $\mathbf{s}_r^i$ is defined as

$$[s_{gr}^{(i,1)}, \ldots, s_{gr}^{(i,M)}, s_{gr}^{(1,i)}, \ldots, s_{gr}^{(M,i)}]^\top.$$

We then compute the Pearson correlation coefficients of these variables, and the results is shown in Table 4.

[6] https://codalab.lisn.upsaclay.fr/competitions/365

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
The limitations section after conclusion

☑ A2. Did you discuss any potential risks of your work?
The ethics statement section after conclusion

☑ A3. Do the abstract and introduction summarize the paper's main claims?
The abstract and section 1

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

B ☑ Did you use or create scientific artifacts?

Section 4

☑ B1. Did you cite the creators of artifacts you used?
Section 4

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
The ethics statement section after conclusion

☑ 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 and section 4

☑ 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?
The ethics statement section after conclusion

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 3

☑ 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 4.1

C ☑ Did you run computational experiments?

Section 4

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

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 4.1 and Appendix A

☑ 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?
Appendix A

☑ 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 4.1 and Appendix A

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.