FCDS: Fusing Constituency and Dependency Syntax into Document-Level Relation Extraction

Xudong Zhu, Zhao Kang*, Bei Hui
University of Electronic Science and Technology of China, Sichuan, China
{2020080902004, zkang, bhui}@uestc.edu.cn

Abstract

Document-level Relation Extraction (DocRE) aims to identify relation labels between entities within a single document. It requires handling several sentences and reasoning over them. State-of-the-art DocRE methods use a graph structure to connect entities across the document to capture dependency syntax information. However, this is insufficient to fully exploit the rich syntax information in the document. In this work, we propose to fuse constituency and dependency syntax into DocRE. It uses constituency syntax to aggregate the whole sentence information and select the instructive sentences for the pairs of targets. It exploits the dependency syntax in a graph structure with constituency syntax enhancement and chooses the path between entity pairs based on the dependency graph. The experimental results on datasets from various domains demonstrate the effectiveness of the proposed method. The code is publicly available at https://github.com/xzAscC/FCDS.

Keywords: Information Extraction, Dependency Graph, Constituency Tree

1. Introduction

Relation Extraction (RE) is a crucial task in information extraction that aims to model relational patterns between entities in an unstructured text. There are two specific scenarios: sentence-level RE and document-level RE. Unlike sentence-level RE (Dixit and Al-Onaizan, 2019; Lyu and Chen, 2021), where entities are located in the same sentence, document-level RE (DocRE) identifies the relation labels between entities within a document. Therefore, DocRE better meets practical needs and has recently received increasing attention (Zhou et al., 2021; Zhao et al., 2022).

A formidable obstacle confronting DocRE is inferring relations of entity pairs in long sentences, which often contain irrelevant or even noisy information (Gupta et al., 2019). Figure 1 is an example, which includes a sentence-level relation and a document-level relation from DocRED. To infer the relation between Louis Chollet and Conservatoire de Paris, models should exclude the influence of unrelated entities and figure out that the word ‘He’ in sentence[2] refers to ‘Louis Chollet’. Buried under massive irrelevant information, DocRE models often struggle with intricate relation instances. Therefore, implicitly learning an instructive context is not sufficient for DocRE (Bai et al., 2021).

In a document, interactions between entities are complex. Pre-trained language models (PLMs) (Kenton and Toutanova, 2019) have shown great potential in many downstream tasks. Some work (Ye et al., 2020; Zhou et al., 2021) implicitly captures such interactions between entities through PLMs. Others, however, model this information explicitly. They first construct document graphs (Zeng et al., 2020; Liu et al., 2023) that consist of different nodes (e.g., mentions, entities, sentences, or the document) to turn instructive context into graphs. Since syntax information can help DocRE by providing explicit syntax refinement and subsentence modeling (Duan et al., 2022), recent studies (Sahu et al., 2019; Wei and Li, 2022) adopt a dependency graph to incorporate both syntax information and structural context. They find that a structural graph can better capture relations and shorten the distance between entities. However, as pointed out in (Sundararaman et al., 2019; Bai et al., 2021), although PLMs are trained with massive real-world text data, there is still a great gap between the implicitly learned syntax and the golden syntax. In fact, syntax information is widely used in sentence-level RE (Xu et al., 2016; Qin et al., 2021), but it has not yet been fully explored under the DocRE scenario.

To fully exploit the syntax information in the document, we fuse constituency and dependency syntaxes in this paper. We mainly adopt dependency graphs and constituency trees to incorporate extra syntax information, and use information from the constituency tree to further enhance the representation of the dependency graph. The dependency and constituency syntax depict complementary but different aspects of syntax information. The dependency graph in figure 2a is mainly used to integrate syntactic information within a single sentence that strongly complements the original plain text, while the constituency tree in figure 2b organizes different words of a single sentence hierarchically and reasonably.

We observe that dependency syntax is better

*Corresponding author

LREC-COLING 2024, pages 7141–7152

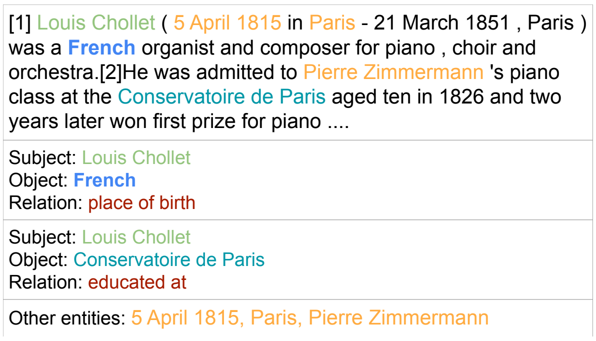


Figure 1: A sentence-level and a document-level relation instance from DocRED. Entity pairs are colored differently according to relation. To identify irrelevant or even noisy relations, unrelated entities are uniformly labeled with one single *color*.

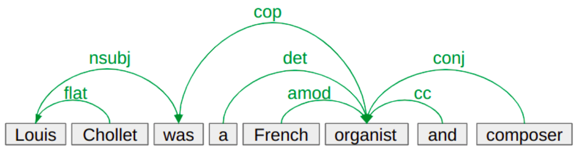
for constructing paths between entity pairs (Wei and Li, 2022), while constituency syntax is better for aggregating sentence-level information. Therefore, we follow previous studies by transforming the dependency tree into a graph and extracting the paths between entity pairs. However, the gap between the learned syntax in PLMs and the golden syntax for dependency trees has heavily influenced the performance of DocRE. To address this issue, we propose to utilize the constituency tree to aggregate sentence-level information to compensate for the gap in the dependency tree. Specifically, we utilize a single-layer MLP to fuse the sentence root in the constituency tree and the dependency graph to replace the original sentence root in the dependency graph. Furthermore, in order to better consider the sentence interaction in the dependency graph, we add a document-level node and link each sentence root to reduce the distance of entity pairs and better capture long-distance relations. Through extensive experiments on three public DocRE benchmarks, DocRED (Yao et al., 2019), CDR (Li et al., 2016), and GDA (Wu et al., 2019), we demonstrate that our model outperforms existing methods.

Our key contributions in this work can be summarized as follows:

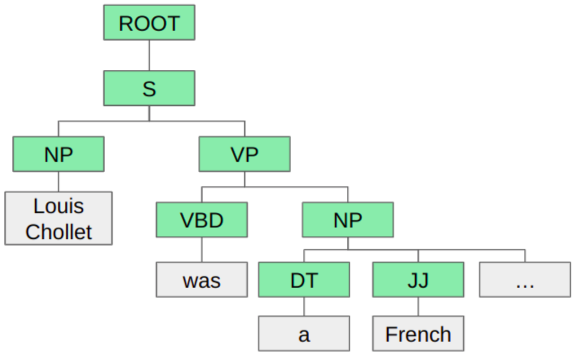
1. We propose to utilize the constituency tree to aggregate sentence-level information to compensate for the gap in the dependency tree and improve the dependency graph by adding a document node to reduce the distance of entity pairs and simplify long-sentence interaction.

2. We process the dependency graph and the constituency tree with Tree-LSTM and GCN, respectively, and set a learnable parameter to adjust their weights.

3. The results of the experiments demonstrate that our model outperforms the existing methods on three DocRE benchmarks, especially on DocRED, where our model improves the IgnF1 of the state-of-the-art methods by at least 1.56%.



(a) Dependency tree describes dependencies between words within a single sentence. Exploiting such syntax information can significantly complement the original plain text and capture the couplings among neighbors. Our model converts the dependency tree to a dependency graph.



(b) Constituency tree organizes the sentence in a tree structure, which not only induces extra hierarchical syntax information but also enables exploring subsentences in arbitrary granularity. Non-leaf nodes are colored green.

Figure 2: Syntactic parsing results of evidence sentence "*Louis Chollet ...*" mentioned in the previous relation instance. (a) and (b) represent the corresponding dependency and constituency tree, respectively.

2. Relation Works

2.1. Document-level Relation Extraction

DocRE is a challenging task since long sentence learning usually requires effective long-distance feature encoding and reasoning (Sahu et al., 2019). To tackle this challenge, some methods apply PLMs for more informative contextual token encoding. (Tang et al., 2020) proposes a hierarchical inference network from the level of entities, sentences, and documents using BERT, while (Ye et al., 2020) explicitly encodes the coreference information to improve the coreferential reasoning ability of BERT. (Xie et al., 2022) empowers DocRE by efficiently extracting evidence and effectively integrating the extracted evidence in inference.

In addition to BERT-based methods, another line of research proposes to use the graph structure to shorten the distances between entities in the document. (Zeng et al., 2020) uses two graphs to represent mention-level and entity-level relations, respectively, while (Wei and Li, 2022) employs linguistic tools to build various edges, such as coreference edges, which embed inter-sentence and intra-sentence dependencies. (Xu et al., 2021b) enforces the model to reconstruct reasoning paths while identifying correct relations. (Duan et al., 2022) utilizes the constituency tree to obtain evidence for DocRE and incorporates dependency graphs to classify the relations.

However, these methods either use regular graph structures that cannot capture sequential information in the original text (Zeng et al., 2020; Xu et al., 2021b), or use dependency and constituency information separately (Wei and Li, 2022; Duan et al., 2022). This work overcomes this drawback by incorporating both constituency and dependency information and enhancing the dependency graph with a constituency tree. With different syntax incorporated, our model can fuse dual-granularity information and better capture long-distance relations.

2.2. Constituency and Dependency Syntax

Since syntax intuitively shares many common features with RE, syntactic features are a highly effective DocRE performance enhancer, according to several empirical verifications in previous work (Zeng et al., 2020; Wei and Li, 2022; Xie et al., 2022). In particular, dependency syntax is extensively studied in DocRE, while constituency syntax is overlooked.

Although the constituency and dependency syntaxes share some common syntactic information, they characterize it from different perspectives. Some work has revealed the mutual benefits of integrating these two heterogeneous syntactic representations for various NLP tasks. (Zhou and Zhao, 2019; Strzyz et al., 2019) integrate dependency and constituency syntactic information as a representation of a parse tree or sequence. (Fei et al., 2021), which is designed for the Semantic Role Labeling (SRL) task, converts the dependency and constituency trees into graphs and performs the graph learning strategy on them. (Dong et al., 2022) proposes to map phrase-level relations in the constituency tree into word-level relations and adopts multi-view learning to capture multiple relationships from the constituency graph and dependency graph for the Open Information Extraction (OpenIE) task, which is the most relevant model to ours.

Our model differs from (Dong et al., 2022) mainly in two aspects: **(1)** (Dong et al., 2022) turns the constituency tree into a graph by heuristic rules and aligns instances of the same node across the dependency graph and constituency graph. Our model, however, utilizes Tree-LSTM to handle the constituency syntax in tree form and compensate for the inaccuracy of dependency; **(2)** To link different syntax, we utilize the constituency tree to enhance the dependency graph instead of adopting multi-view learning to fuse heterogeneous information from both graphs.

3. Methodology

A document $\mathcal{D}$ contains I sentences $\{sen_i\}_{i=1}^{I}$ and N entities $\{e_i\}_{i=1}^{N}$. sen_i is the i^{th} sentence, which includes P_i tokens: $\{t_{i,1}, t_{i,2}, \cdots, t_{i,P_i}\}$. An entity e_k can have Q_k mentions $\{m_{k,1}, m_{k,2}, \ldots, m_{k,Q_k}\}$.

The goal of DocRE is to correctly infer all relations between each entity pair $(e_s, e_o)_{s,o=1,2,\cdots,N;s\neq o}$, where e_s is a subject entity and e_o is an object entity. The predicted relations are subsets of the predefined relation set $\mathcal{R}$ or $\{\mathcal{NA}\}$ (without relation).

The overall architecture of the Fusing Constituency and Dependency Syntax (FCDS) is illustrated in Figure 3. We exploit dependency and constituency syntax to build a dependency graph and constituency tree and utilize BERT to encode words in the document. Then we use Tree-LSTM to aggregate information from the constituency tree and exploit the dependency graph by graph neural network (GNN) (Kipf and Welling, 2017; Dai et al., 2022) while utilizing the constituency tree to improve the dependency graph. We obtain relations between entity pairs by a learnable weight to combine the dependency graph and the constituency tree.

3.1. Text Encoding

Given a document, a special marker '*' (Zhang et al., 2017) will be first inserted before and after each mention. Then we feed tokens from document $\mathcal{D}$ into PLMs to obtain contextualized representation $\mathcal{H} = \{h_1, \ldots, h_T\} \in \mathcal{R}^{T\times d}$, where T is the number of tokens and d is the dimension of token embedding:

$$\mathcal{H} = PLM\{x_{1,1}, \cdots, x_{I,P_I}\} \quad (1)$$

$x_{i,j}$ is the j^{th} word for the i^{th} sentence in the document and the PLMs can be a pre-trained BERT (Kenton and Toutanova, 2019) or LSTM model.

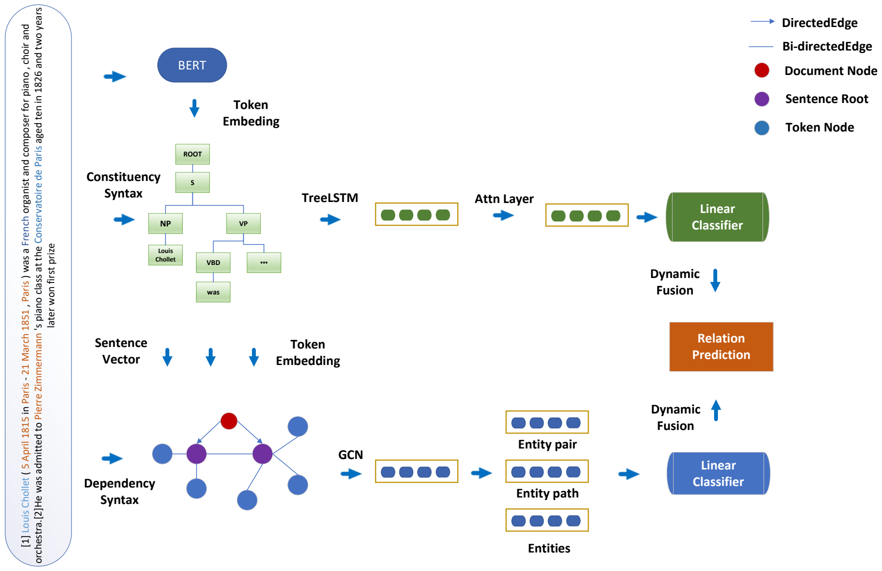


Figure 3: The overview of our architecture. Note that we use the result of constituency syntax to enhance the dependency graph and obtain relations between entity pairs with dynamic weighted fusion.

3.2. Constituency Tree Construction

After obtaining the token-level embedding through the PLMs model, FCDS utilizes a constituency tree to compensate for the inaccuracy in the dependency tree based on hierarchical syntax information. Specifically, we take advantage of the constituency tree based on the document $\mathcal{D}$ and use Tree-LSTM (Miwa and Bansal, 2016; Duan et al., 2022) to incorporate root information of a sentence. First, we parse each sentence into the corresponding constituency tree[1]. We can see from Figure 2b that each constituency tree describes a logical way to restore the entire sentence piece by piece. Using a constituency tree, we can not only incorporate extra hierarchical syntax information but also encode sentences with arbitrary granularity.

To represent the constituency tree with Tree-LSTM, we first initialize hidden state h_j and memory cell state c_j with zeros. The input vector of leaf nodes is initialized with their corresponding representations inside PLMs, while non-leaf nodes are set to zeros. We then broadcast features of leaf nodes all the way up to the root node using Tree-LSTM (Miwa and Bansal, 2016). The input gate i_j, the output gate o_j, and the forget gate f_{jk} of an arbitrary node j in the constituency tree are calculated as:

$$i_j = \sigma(W_i x_j + \sum_{l \in Child(j)} h_l W_{il} + b_i) \quad (2)$$

$$o_j = \sigma(W_o x_j + \sum_{l \in Child(j)} h_l W_{ol} + b_o) \quad (3)$$

[1] constituency trees are obtained using the Stanza library. https://stanfordnlp.github.io/stanza/

$$f_{jk} = \sigma(W_f x_j + \sum_{l \in Child(k)} h_{jl} W_{kl} + b_f) \quad (4)$$

where σ is the sigmoid function, W, b are trainable parameters. x_j denotes the input vector of node j and $Child(j)$ means the child of node j in the constituency tree.

The integrated result u_j is calculated as:

$$u_j = \tanh(W_u x_j + \sum_{l \in Child(j)} h_l W_{ul} + b_u) \quad (5)$$

At last, we update hidden state h_j and memory cell state c_j as follow,

$$c_j = i_j \odot u_j + \sum_{l \in Child(j)} f_{jl} \odot c_l \quad (6)$$

$$h_j = o_j \odot \tanh(c_j) \quad (7)$$

In practice, we extract information from its child nodes recurrently and use the sentence root feature as the sentence feature vector. For a document with I sentences, we can get all the sentence vectors $\{s_i\}_{i=1}^{I}$. Since not all sentences contain relevant information for relation reasoning, we employ a multi-head attention layer over the sentence vector to identify the most relevant ones. We formulate this process as:

$$V_{doc} = [s_1, s_2, \ldots, s_I] \quad (8)$$

$$S,\ A = Attn(W_{t1}(e_s - e_o), W_{t2} V_{doc}, W_{t3} V_{doc}) \quad (9)$$

where W_{t1}, W_{t2}, and W_{t3} are trainable weights and $Attn$ represents an attention layer. A is the attention score and S is the sentence vector through the attention layer. Then, we combine the weighted sentence with entity pairs and calculate the score z_{const} of relation r:

$$z_s = tanh(W_{(s1)}e_s + W_{(s2)}S) \tag{10}$$

$$z_o = tanh(W_{(o1)}e_o + W_{(o2)}S) \tag{11}$$

$$z_{const} = z_s^{\top} W_{s,o} z_o + b_{const} \tag{12}$$

where W, b are trainable parameters. As a result, we obtain the relation scores z_{const} based on the constituency tree with attention score A and sentence vector S to enhance the dependency graph.

3.3. Dependency Graph Construction

After obtaining the above representation, we construct a dependency graph to aggregate information of syntactically associated words. Each sentence in the document is fed into a dependency parser, which generates a dependency syntax tree. Then we convert the dependency trees to dependency graphs. Note that we add a document node to shorten the distance between entity pairs and better aggregate information of long-distance sentences.

The dependency graph contains four kinds of nodes: non-root token nodes, root token nodes, mention nodes, and document nodes. Specifically, each token in the document corresponds to a token node, and for tokens that are not the sentence root in the dependency tree, its encoded feature corresponds to its node feature. For sentence root, we treat it especially to combat possible error parsing, taking advantage of the root feature S obtained by the Eq. 9. We use MLP to fuse the original feature with S in Eq. 9 since the sentence root in the dependency graph is expected to include information of the entire sentence. For the mention node, the node feature is calculated by averaging the features of tokens in this mention. The document node, as a node that includes information of each sentence root and emphasizes the conducive sentences, is calculated as the weighted average of each sentence vector. The weight is the attention score of the entity pairs in Eq. 9 .

There are four types of edges in this graph. Three of them are bi-directed, and one is directed. Bi-directed dependency syntax edges are added between each pair of connected tokens in the syntax tree. Then the bi-directed edges are added between the dependency syntax tree roots of adjacent sentences, since there exist close context relationships between adjacent sentences. As each sentence in the document serves the same topic of this document, bi-directed edges are added between dependency syntax tree roots and the document node. The last type of edge is directed and exists between non-adjacent sentence roots to capture long-distance information and embed sequential information.

The weight of bi-directed edges is 1 to inform their strong connection. The weight $ADJ_{i,j}$ of directed edges from non-adjacent sentence root node i to j are calculated based on their feature vectors:

$$ADJ_{i,j} = \frac{S_i \cdot S_j}{||S_i|| \cdot ||S_j||} \tag{13}$$

where S_i and S_j are the sentence vector i and j in Eq. 9. Using these learned weights, our model can obtain a logic flow from the earlier root to the later root automatically and fuse the information between dependency and constituency syntax to aggregate information from different perspectives.

After obtaining the adjacent matrix ADJ of the dependency graph, we employ GCN for feature aggregation and entity strengthening.

$$q^{l+1} = GCN(q^l, ADJ) \tag{14}$$

where q^l is the input feature of layer l and q^{l+1} is the output feature.

Then entity representation $entity_i$ is abstracted by merging the embeddings of all mentions of this entity based on logsumexp (Jia et al., 2019):

$$entity_i = log \sum_j exp(m_{i,j}) \tag{15}$$

where $m_{i,j}$ is j^{th} mention embedding of $entity_i$.

In addition to entity-specific embeddings, we also extract the shortest path $path_{s,o}$ between two targeted entities to complement the entity pair (e_s, e_o) on the dependency graph.

$$path_{s,o} = [e_s, node_1, node_2, \cdots, node_n, e_o] \tag{16}$$

For efficiency, we limit the maximum length of the selected path to 14, which means that a maximum of 12 path nodes will be selected except the head and tail entities. Any path longer than 12 will select the first 12 nodes, while a path less than 12 will be filled with zero tensor. In addition, we use MLP to explore the relation of entity pairs.

$$pair_{s,o} = LeakyRelu(W_{p1}e_s + W_{p2}e_o) \tag{17}$$

Through the above steps, dependency syntax complemented entity and context representations are acquired. Following the previous methods (Mou et al., 2016), we then concatenate them all to strengthen the features of this entity pair:

$$I_{s,o} = [e_s; e_o; pair_{s,o}; path_{s,o}] \tag{18}$$

We compute the relation score z_{dep} based on dependency graph:

$$z_{dep} = W_{(d2)}\sigma(W_{(d1)}I_{s,o} + b_{(d1)}) + b_{(d2)} \quad (19)$$

where W, b are trainable parameters.

3.4. Dynamic Fusion and Classification

Finally, we combine two scores acquired by the dependency graph and constituency tree as a dynamic weighted sum of them (Kendall et al., 2018).

$$z_{final} = z_{dep} + \eta z_{const} \quad (20)$$

We adopt adaptive margin loss as loss function (Wei and Li, 2022).

$$\mathcal{L} = \sum_{1 \leq i \leq C} max(0, \alpha - c_i(z^i_{final} - z^s_{final})) \quad (21)$$

where $\alpha > 0$ is a hyper-parameter for margin and c_i is 1 if the sample belongs to the positive class and -1 otherwise. C is the number of classes and z^i_{final}, z^s_{final} is the final score of $non\mathcal{NA}, \mathcal{NA}$ classes. Note that the proposed adaptive margin loss is reduced to Hinge loss (Gentile and Warmuth, 1998) in the binary RE tasks.

4. Experiments

4.1. Datasets

To comprehensively evaluate our model, we assess the proposed model on three document-level datasets from various domains. Statistics of these datasets are listed in Table 1.

- **DocRED** (Yao et al., 2019) is a large-scale human-annotated dataset constructed from Wikipedia and Wikidata. It contains 132,275 entities, 56,354 relational facts and 96 relation classes. More than 40.7% of the relation pairs are cross-sentence relation facts.

- **CDR** (Li et al., 2016) is a biomedical DocRE dataset built from 1,500 PubMed abstracts that are randomized into three equal parts for training, validation and testing. The task is to predict the binary relation between Chemicals and Diseases.

- **GDA** (Wu et al., 2019) is also a biomedical DocRE dataset contains 30,192 abstracts. The dataset is annotated with binary relations between Gene and Disease concepts using distant supervision.

Table 1: Statistics of three benchmarks used in our experiments.

Statistics	DocRED	CDR	GDA
# Train	3053	500	23353
# Dev	1000	500	5839
# Test	1000	500	1000
# Relations	96	2	2
Avg.# sentences per Doc	8.0	9.7	10.2

4.2. Implementation details

Our model is implemented on Pytorch (Paszke et al., 2019) and uses stanza (Qi et al., 2020) to extract constituency and dependency syntax. For all experiments, the learning rate is set to 5e-5 and the weight decay is 1e-4. The GCN layer number for the dependency graph is set to 3 and the output dimension is 128. The hidden state and cell state of each node in the constituency tree share a dimension of 256. α in Eq. 21 is set to 1.0 and learning rate warmup (Goyal et al., 2017) with ratio 0.06 is implemented followed by a linear decay to 0. The entire model is optimized by AdamW (Loshchilov and Hutter, 2019).

4.3. Results on DocRED

We compare our model with graph-based methods and BERT-based methods in DocRED. For BERT-based methods, we compare the proposed method with BERT (Yao et al., 2019), ATLOP (Zhou et al., 2021), evidence-based EIDER (Xie et al., 2022), and self-training method DREEAM (Ma et al., 2023). Graph-based models include LSR (Nan et al., 2020), HeterGSAN (Xu et al., 2021b), DRE (Xu et al., 2021a), CorefDRE (Xue et al., 2022), GAIN (Zeng et al., 2020), SagDRE (Wei and Li, 2022), and LARSON (Duan et al., 2022). Following previous work (Zhou et al., 2021), we train our model on BERT_{base} and $\mathrm{DeBERTa}_{Large}$. We report not only F1 and Ign F1 (F1 score excluding the relational facts shared by the training and dev/test set) following the prior studies (Yao et al., 2019), but also Intra F1 (F1 that only considers intra-sentence relational facts) and Inter F1 (F1 that only considers inter-sentence relational facts).

The experimental results listed in Table 2 show that our model can achieve leading performance in DocRE data in the general domain. The proposed model outperforms the dependency graph-based methods SagDRE (Wei and Li, 2022) by margins of 2.29% and 1.94% on the test set in terms of Ign F1 and F1, respectively, indicating that the combination of dependency and constituency syntax is useful for document-level relation extraction. Our model improves the IgnF1 score on the test set by 1.56% over the state-of-the-art method LARSON (Duan et al., 2022), which uses the constituency

Table 2: Results (%) of relation extraction on the dev and test set of DocRED. Results of other methods are directly taken from original papers.

Model	dev				test	
	IgnF1	F1	Intra F1	Inter F1	IgnF1	F1
BERT_{base}(Yao et al., 2019)	-	54.16	61.61	47.15	-	53.20
$\mathrm{LSR-BERT}_{base}$(Nan et al., 2020)	52.43	59.00	65.26	52.05	56.97	59.05
$\mathrm{GAIN-BERT}_{base}$(Zeng et al., 2020)	59.14	61.22	67.10	53.90	59.00	61.24
$\mathrm{HeterGSAN-BERT}_{base}$(Xu et al., 2021b)	58.13	60.18	-	-	57.12	59.45
$\mathrm{DRN-BERT}_{base}$(Xu et al., 2021a)	59.33	61.09	-	-	59.15	61.37
$\mathrm{ATLOP-BERT}_{base}$(Zhou et al., 2021)	59.22	61.09	-	-	59.31	61.30
$\mathrm{CorefDRE-BERT}_{base}$(Xue et al., 2022)	60.85	63.06	-	-	60.78	60.82
$\mathrm{EIDER-BERT}_{base}$(Xie et al., 2022)	60.51	62.48	68.47	55.21	60.42	62.47
$\mathrm{SagDRE-BERT}_{base}$(Wei and Li, 2022)	60.32	62.06	-	-	60.11	62.32
$\mathrm{LARSON-BERT}_{base}$(Duan et al., 2022)	61.05	63.01	68.63	55.75	60.71	62.83
$\mathrm{DREEAM-BERT}_{base}$(Ma et al., 2023)	60.51	62.55	-	-	60.03	62.49
FCDS-BERT$_{base}$	**62.61**	**64.42**	**68.79**	**57.24**	**62.08**	**64.21**
$\mathrm{ATLOP-DeBERTa}_{Large}$(Zhou et al., 2021)	62.16	64.01	68.45	59.63	62.12	64.08
FCDS-DeBERTa$_{Large}$	**64.12**	**66.17**	**70.19**	**58.73**	**64.03**	**65.86**

Table 3: F1 Results (%) of relation extraction on the test set of CDR and GDA.

Model	CDR	GDA
LSR-BERT(Nan et al., 2020)	64.80	82.20
SciBERT(Zhou et al., 2021)	65.10	82.50
ATLOP-SciBERT(Zhou et al., 2021)	69.40	83.90
EIDER-SciBERT(Xie et al., 2022)	70.63	84.54
SagDRE-SciBERT(Wei and Li, 2022)	71.80	-
LARSON-SciBERT(Duan et al., 2022)	71.59	86.02
DREEAM-SciBERT(Ma et al., 2023)	71.55	84.51
FCDS-SciBERT	**72.62**	**87.39**

tree to predict the evidence for DocRE and uses the constituency and dependency syntax separately. The advance confirms that information of different granularity can assist relation extraction in DocRE.

4.4. Results on Biomedical Datasets

In addition to general domain DocRE methods, we also compare our model with various advanced methods including LSR (Nan et al., 2020), sciBERT (Zhou et al., 2021), ATLOP (Zhou et al., 2021), EIDER (Xie et al., 2022), SagDRE (Wei and Li, 2022) and LARSON (Duan et al., 2022) on two biomedical domain datasets CDR and GDA.

Experimental results are listed in Table 3. In summary, our model achieves significant improvements over two tested datasets (0.93% on CDR and 1.27% on GDA). GDA and CDR have more sentences than DocRED, thus our model can deal with complex documents, further demonstrating its generality.

Furthermore, to assess the significance of improvements, we perform a two-sample t-test comparing our approach with five other methods in three datasets. The obtained p-values are presented in Table 4. It can be seen that all values are less than 0.05, demonstrating a significant improvement of FCDS.

Table 4: Two-sample t-test on all datasets. Use BERT for embeddings on DocRED, and SciBERT on CDR and GDA.

Model \ Metric	IgnF1	F1
LSR(Nan et al., 2020)	0.0146	0.0129
ATLOP(Zhou et al., 2021)	0.0240	0.0213
EIDER(Xie et al., 2022)	0.0337	0.0313
LARSON(Duan et al., 2022)	0.0352	0.0341
DREEAM(Ma et al., 2023)	0.0470	0.0439

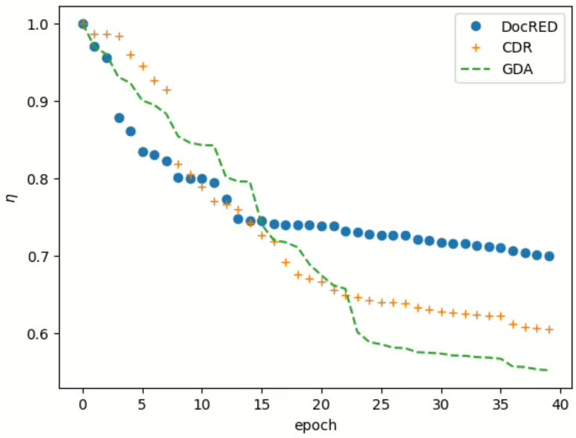


Figure 4: The learning curve of η on DocRED, CDR and GDA datasets.

Table 5: Ablation study of our model in dev set of DocRED.

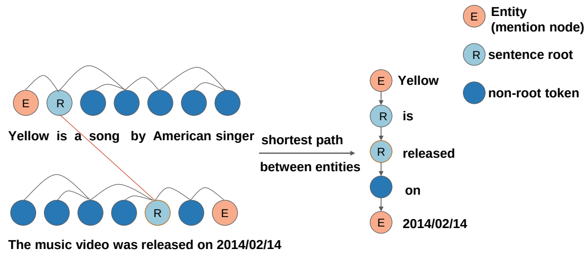
Ablation	F1	IgnF1
FCDS	64.42	62.61
separate syntax	62.59	60.63
w/o document node	62.78	60.92
w/o constituency tree	62.15	60.22
w/o dependency graph	61.63	59.41

4.5. Ablation Study

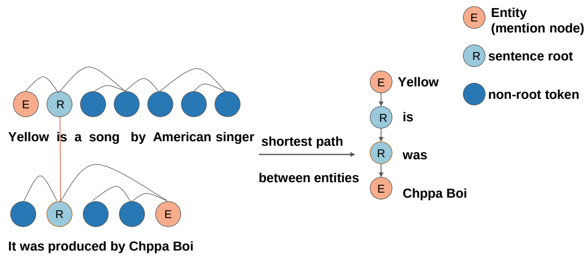
To exhaustively understand how each component of our method contributes to the final performance, we perform ablation studies to analyze the function of different syntaxes. In Figure 4, we plot the learning curve of η in Eq. 20. We can observe that η decreases from the initial value of 1.0, indicating that the dependency graph plays a more vital role during the process, possibly due to the fact that the dependency graph combines the features of the constituency tree and captures fine-granularity information.

Furthermore, we remove one component at a time and assess the resulting model using the dev set in DocRED in Table 5. For separate syntax case, we do not use the constituency tree to enhance the dependency graph and observe that F1 and IgnF1 decrease by 0.78% and 0.94%, indicating that fusing syntax is beneficial for DocRE. Then we remove the document node from the dependency graph, dependency graph, and constituency tree, respectively. For w/o document node, we observe that F1 and IgnF1 decrease by 0.59% and 0.65%. We speculate that the document node reduces the distance between entity pairs, which contributes to DocRE. For the w/o dependency graph, only the constituency tree is incorporated for relation prediction. We can observe that the F1 score decreases by 2. 16%. Similar trends occur when we remove the constituency tree, where F1 decreases to 62.15%. Therefore, both dependency graphs and constituency trees are significant for our models.

Finally, to examine the impact of document nodes on reducing the distance between entity pairs, we analyze the changes in entity distances before and after adding document-level nodes in three datasets in Table 6. Through a random selection of 600 cases, we assess the average, maximum, minimum, and standard deviation of entity distances with and without document nodes. Our finding indicates a reduction in average entity distances, suggesting that document nodes could improve DocRE by simplifying interactions between entities.



(a) In this case, the dependency graph plays a positive role in DocRE by extracting the shortest path '**Yellow** is released on **2014/02/14**' between entities **Yellow** and **2014/02/14**.



(b) In this case, the dependency graph plays an ambivalent role in DocRE. It extracts the shortest path '**Yellow** is was **Chppa Boi**' between **Yellow** and its author **Chppa Boi**. However, due to the inaccuracy of the dependency parser, the model cannot capture the keyword **produced**, therefore it is hard to predict the real relation between entities.

Figure 5: Two different cases in DocRED.

4.6. Case Study

To better understand the bottleneck of FCDS and inspire future work, we conduct a case study to investigate the predictions that FCDS makes. The result is shown in Figure 5.

The first case in Figure 5a is a successful dependency graph and illustrates how the dependency syntax helps the model complete DocRE. In this case, the dependency parser successfully parses the dependency syntax within two sentences and extracts vital keywords to manage the prediction. With the short path "Yellow is released on 2014/02/14.", simple sentence-level models can finish the prediction.

The second case in Figure 5b illustrates the motivation behind our methods, which is to alleviate the inaccuracy of the dependency parser and the failure of selecting keywords for entities. Although the dependency syntax is a useful tool for DocRE, it sometimes fails to identify the relevant keywords for entities. In this case, the shortest path "Yellow

	DocRED		CDR		GDA	
	with document node	w/o document node	with document node	w/o document node	with document node	w/o document node
avg & std	6.19 ± 1.02	7.23 ± 1.55	5.96 ± 2.26	6.44 ± 2.23	6.82 ± 1.99	7.38 ± 2.43
max	7	9	8	12	11	14
min	4	4	4	4	3	4

Table 6: The average/max/min length and standard deviation of entity distances with document node and w/o document node in DocRED, CDR, GDA.

is was Chppa Boi" is far from finishing the prediction, while the real keyword "produced" is not selected in the path. Furthermore, errors in the dependency parser lead to a severe lack of information. To address this issue, we utilize the constituency tree and fuse the information from dependency and constituency syntax, which organizes different words of a single sentence hierarchically and can aggregate the sentence-level information naturally towards the sentence root. By doing so, the sentence root can obtain the mixed information of the root itself and the sentence information, adding the information of the path and alleviating the inaccuracy of the dependency parser.

5. Conclusion

In this work, we propose a novel model for the document-level relation extraction task. Our model exploits two types of extra syntax information, namely dependency syntax and constituency syntax. GCN and Tree-LSTM are adopted to encode the two types of information. Furthermore, by using the constituency tree to enhance the dependency graph and adding a document node in the dependency graph, we can improve the expression capability of the dependency graph and better capture long-distance correlations. Experiments on three public DocRE datasets demonstrate that our model outperforms the existing method. In the future, we plan to select the most conducive sentences for entity pair by constituency tree, which captures the information from another perspective and complements the dependency graph.

6. Acknowledgements

This work was supported by the National Natural Science Foundation of China (Nos. 62276053 and 62273071) and High-performance Computing Platform of UESTC.

7. Bibliographical References

Jiangang Bai, Yujing Wang, Yiren Chen, Yaming Yang, Jing Bai, Jing Yu, and Yunhai Tong. 2021. Syntax-BERT: Improving pre-trained transformers with syntax trees. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 3011–3020, Online. Association for Computational Linguistics.

Yong Dai, Linjun Shou, Ming Gong, Xiaolin Xia, Zhao Kang, Zenglin Xu, and Daxin Jiang. 2022. Graph fusion network for text classification. *Knowledge-Based Systems*, 236:107659.

Kalpit Dixit and Yaser Al-Onaizan. 2019. Span-level model for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5308–5314, Florence, Italy. Association for Computational Linguistics.

Kuicai Dong, Aixin Sun, Jung jae Kim, and Xiaoli Li. 2022. Syntactic multi-view learning for open information extraction. In *Conference on Empirical Methods in Natural Language Processing*.

Zhichao Duan, Xiuxing Li, Zhenyu Li, Zhuo Wang, and Jianyong Wang. 2022. Not just plain text! fuel document-level relation extraction with explicit syntax refinement and subsentence modeling. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1941–1951, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Hao Fei, Shengqiong Wu, Yafeng Ren, Fei Li, and Donghong Ji. 2021. Better combine them together! integrating syntactic constituency and dependency representations for semantic role labeling. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 549–559, Online. Association for Computational Linguistics.

Claudio Gentile and Manfred K. K Warmuth. 1998. Linear hinge loss and average margin. In *Advances in Neural Information Processing Systems*, volume 11. MIT Press.

Priya Goyal, Piotr Dollár, Ross B. Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch SGD: training imagenet in 1 hour. *CoRR*, abs/1706.02677.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 6513–6520.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multiscale representation learning. 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 3693–3704, Minneapolis, Minnesota. Association for Computational Linguistics.

Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7482–7491.

Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, volume 1, page 2.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

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.

Hongfei Liu, Zhao Kang, Lizong Zhang, Ling Tian, and Fujun Hua. 2023. Document-level relation extraction with cross-sentence reasoning graph. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 316–328. Springer.

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

Shengfei Lyu and Huanhuan Chen. 2021. Relation classification with entity type restriction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 390–395, Online. Association for Computational Linguistics.

Youmi Ma, An Wang, and Naoaki Okazaki. 2023. Dreeam: Guiding attention with evidence for improving document-level relation extraction. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, EACL, page (to appear), Dubrovnik, Croatia. Association for Computational Linguistics.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1116, Berlin, Germany. Association for Computational Linguistics.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. 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.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A Python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*.

Han Qin, Yuanhe Tian, and Yan Song. 2021. Relation extraction with word graphs from n-grams. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2860–2868, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4309–4316, Florence, Italy. Association for Computational Linguistics.

Michalina Strzyz, David Vilares, and Carlos Gómez-Rodríguez. 2019. Sequence labeling parsing by learning across representations. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5350–5357, Florence, Italy. Association for Computational Linguistics.

Dhanasekar Sundararaman, Vivek Subramanian, Guoyin Wang, Shijing Si, Dinghan Shen, Dong Wang, and Lawrence Carin. 2019. Syntax-infused transformer and bert models for machine translation and natural language understanding. *arXiv preprint arXiv:1911.06156*.

Hengzhu Tang, Yanan Cao, Zhenyu Zhang, Jiangxia Cao, Fang Fang, Shi Wang, and Pengfei Yin. 2020. Hin: Hierarchical inference network for document-level relation extraction. *Advances in Knowledge Discovery and Data Mining*, 12084:197 – 209.

Ying Wei and Qi Li. 2022. Sagdre: Sequence-aware graph-based document-level relation extraction with adaptive margin loss. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 2000–2008, New York, NY, USA. Association for Computing Machinery.

Ye Wu, Ruibang Luo, Henry CM 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: 23rd Annual International Conference, RECOMB 2019, Washington, DC, USA, May 5-8, 2019, Proceedings 23*, pages 272–284. Springer.

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.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021a. Discriminative reasoning for document-level relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1653–1663, Online. Association for Computational Linguistics.

Wang Xu, Kehai Chen, and Tiejun Zhao. 2021b. Document-level relation extraction with reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14167–14175.

Yan Xu, Ran Jia, Lili Mou, Ge Li, Yunchuan Chen, Yangyang Lu, and Zhi Jin. 2016. Improved relation classification by deep recurrent neural networks with data augmentation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1461–1470, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhongxuan Xue, Rongzhen Li, Qizhu Dai, and Zhong Jiang. 2022. Corefdre: Document-level relation extraction with coreference resolution. *arXiv preprint arXiv:2202.10744*.

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.

Deming Ye, Yankai Lin, Jiaju Du, Zhenghao Liu, Peng Li, Maosong Sun, and Zhiyuan Liu. 2020. Coreferential Reasoning Learning for Language Representation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7170–7186, Online. Association for Computational Linguistics.

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.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, Copenhagen, Denmark. Association for Computational Linguistics.

Chao Zhao, Daojian Zeng, Lu Xu, and Jianhua Dai. 2022. Document-level relation extraction with context guided mention integration and inter-pair reasoning. *ArXiv*, abs/2201.04826.

Junru Zhou and Hai Zhao. 2019. Head-Driven Phrase Structure Grammar parsing on Penn Treebank. In *Proceedings of the 57th Annual*

Meeting of the Association for Computational Linguistics, pages 2396–2408, Florence, Italy. 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. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 14612–14620.