Not Just Plain Text! Fuel Document-Level Relation Extraction with Explicit Syntax Refinement and Subsentence Modeling

Zhichao Duan[1], **Xiuxing Li**[2,3]*, **Zhenyu Li**[1], **Zhuo Wang**[1], **Jianyong Wang**[1]*

[1]Department of Computer Science and Technology, Tsinghua University
[2]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[3]University of Chinese Academy of Sciences
{dzc20,zy-li21,wang-z18}@mails.tsinghua.edu.cn
lixiuxing@ict.ac.cn; jianyong@tsinghua.edu.cn

Abstract

Document-level relation extraction (DocRE) aims to identify semantic labels among entities within a single document. One major challenge of DocRE is to dig decisive details regarding a specific entity pair from long text. However, in many cases, only a fraction of text carries required information, even in the manually labeled supporting evidence. To better capture and exploit instructive information, we propose a novel ex**p**Licit synt**A**x **R**efinement and **S**ubsentence m**O**deli**N**g based framework (LARSON). By introducing extra syntactic information, LARSON can model subsentences of arbitrary granularity and efficiently screen instructive ones. Moreover, we incorporate refined syntax into text representations which further improves the performance of LARSON. Experimental results on three benchmark datasets (DocRED, CDR, and GDA) demonstrate that LARSON significantly outperforms existing methods.

1 Introduction

Relation extraction (RE) is an essential task in information extraction. It aims to model relational patterns between entities in unstructured text. One strikingly significant variant of RE, document-level relation extraction (DocRE) is designed to identify relations among entity pairs distributed throughout the document. Compared to traditional sentence-level RE (Dixit and Al-Onaizan, 2019; Lyu and Chen, 2021; Zhou and Chen, 2021), where entities are located in the same sentence, DocRE further fits the need in real scenes and has received increasing attention lately (Christopoulou et al., 2019; Wang et al., 2020; Xu et al., 2021b; Xie et al., 2022; Zhao et al., 2022).

In an ordinary document, interactions between entities are complex. Since pre-trained language models (PLMs) have shown their great potential

---
*Corresponding authors.

[1] During her youth, *Michelle* lived with her grandparents in Northwestern France before enrolling into an international school in Kobe, Japan. [2] Ferre later enrolled in… [3] Her ability to speak both *English* and Japanese led her to pursue a career in journalism. [4] She lived in … [5] Michelle Ferre 's first acting experience was opposite Jackie Chan in the 1998 *Hong Kong* martial arts film *Who Am I* ?. [6] Jackie Chan became intrigued by Ferre's personality while being interviewed on the set of *Who Am I* ? …

Subject: *Michelle*		**Object:** *English*	
Relation: language spoken		**Evidence:** [1, 3]	
Subject: *Hong Kong*		**Object:** *Who Am I*	
Relation: place of origin		**Evidence:** [5]	

Figure 1: We take an inter-sentence and an intra-sentence relation instance from DocRED. Decisive subsentences are underlined with different colors according to relation *language spoken* and *place of origin*. To identify the relation among (*Michelle*, *English*), extra coreference resolution between referring expression *Her* and referent *Michelle* is needed.

in many downstream tasks (Duan et al., 2021; Li et al., 2022), some works implicitly capture such interactions through pre-trained language models (PLMs) (Wang et al., 2019; Yuan et al., 2021; Zhou et al., 2021; Xie et al., 2022). Some other works model this information explicitly. They first construct document graphs that consist of different nodes (e.g., mentions, entities, sentences, or the document) (Christopoulou et al., 2019; Zeng et al., 2020; Wang et al., 2020; Xu et al., 2021a). Then, graph convolutional networks (GCNs) (Kipf and Welling, 2016; Brody et al., 2021) are adopted to encode precise node representations and infer final relations. According to their experiments, GCNs can better capture complex interconnections between nodes which is the foundation of our method.

One major challenge around DocRE concerns inferring relations of multiple entity pairs across long text, which may contain irrelevant or even noisy information. Take Figure 1 as an example. It includes an inter-sentence and an intra-sentence

relation instance. To infer the inter-sentence relation between *Michelle* and *English*, human can quickly locate the subsentence "*Her ability ...*" and deduce that *Her* is referring to *Michelle* here. Furthermore, given (*Hong Kong*, *Who Am I*), human can identify their intra-sentence relation just by a prepositional phrase "*in the 1998 Hong Kong martial arts film Who Am I?* ". Unfortunately, such fine-grained annotations are not available in current DocRE datasets. With massive irrelevant information, DocRE models sometimes struggle with complicated relation instances, indicating that implicitly learning instructive context is not sufficient. Another factor that hinders the further development of DocRE algorithms is syntax information. As pointed out in Sundararaman et al., 2019; Bai et al., 2021, though PLMs are trained with massive real-world text data, there is still a big gap between implicitly learned syntax and the golden syntax. Moreover, syntax information is widely incorporated in many sentence-level RE models (Xu et al., 2016; Zhang et al., 2018; Qin et al., 2021) and is not yet developed sufficiently under DocRE scenario (Gupta et al., 2019).

In order to better solve DocRE, we propose a novel exp**L**icit synt**A**x **R**efinement and **S**ubsentence m**O**deli**N**g based framework (LARSON). We mainly exploit dependency and constituency trees to incorporate extra syntax information and model subsentences. The two trees depict complete yet different aspects of syntax information. For example, the evidence sentence as in the previous intra-sentence relation instance can be parsed into corresponding dependency tree and constituency tree[1]. We can see from Figure 2a that a dependency tree describes the dependencies between words within a single sentence that strongly complement the original plain text. Moreover, it facilities coupling syntactically associated words. The constituency tree shown in Figure 2b can organize different words of a single sentence hierarchically and reasonably, eliminating the process of enumerating different word combinations while maintaining hierarchical syntax information. With the aid of dedicated attention module (Bahdanau et al., 2014), we can even gather all subsentences with proper weights to resolve inter-sentence dependencies and produce final relations. To the best of our knowledge, LARSON is the first DocRE model that con-
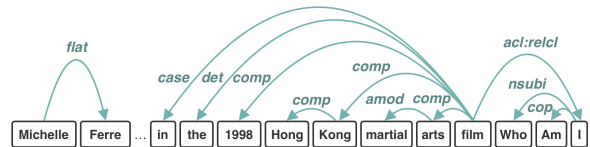
(a) Dependency tree describes dependencies between words within a single sentence. Exploiting such syntax information can significantly complement the original plain text and facilitate coupling neighbor information.
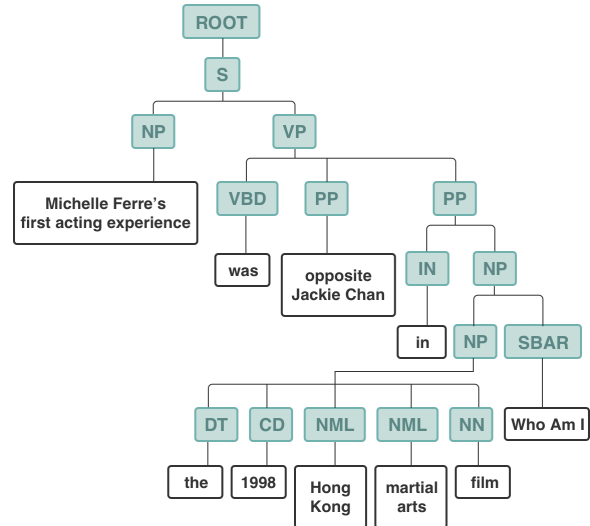
(b) Constituency tree organizes the sentence in a tree structure which not only induces extra hierarchical syntax information but also enables exploring subsentences of arbitrary granularity.

Figure 2: Syntactic parsing results of evidence sentence "*Michelle Ferre ...*" mentioned in the previous intra-sentence instance. (a) and (b) represent the corresponding dependency and constituency tree, respectively. Irrelevant nodes (words) are either collapsed or neglected.

siders subsentence modeling. Through extensive experiments on three public DocRE benchmarks, DocRED (Yao et al., 2019), CDR (Li et al., 2016), and GDA (Wu et al., 2019b), we demonstrate that our model LARSON outperforms existing methods by a large margin.

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

1. We propose enhancing text representations through dependency trees to complement the original plain text and aggregate information of syntactically associated words.

2. We propose encoding subsentences through constituency trees to help LARSON focus more on valuable pieces while maintaining hierarchical syntax information.

---

[1]Dependency and constituency trees are obtained using the Stanza library. https://stanfordnlp.github.io/stanza/
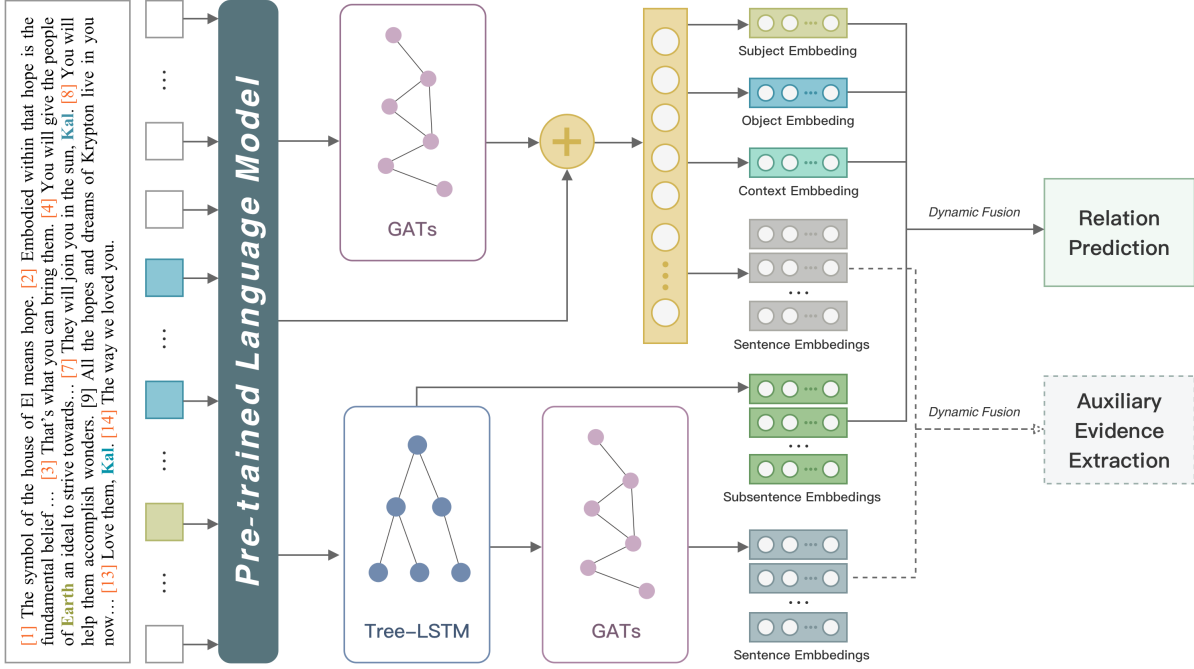
Figure 3: The overall architecture of LARSON. Note that the two GATs share the same architecture but have different parameters.

3. Experimental results demonstrate that LAR-SON achieves leading performance on all three DocRE benchmarks, especially on GDA, where our model outperforms state-of-the-art method by 1.48% F_1.

2 Problem Formulation

Given a document D made up of sentences $\{sen_i\}_{i=1}^I$ and appearing entities $\{e_i\}_{i=1}^N$, the goal of DocRE is to correctly infer all the existing relations between each entity pair $(e_s, e_o)_{s,o=1...N;s\neq o}$ where e_s is the subject entity and e_o is the object entity. Among which, entity e_i usually consists of multiple mentions and the jth mention of e_i is denoted as $m_{(i,j)}$. Predicted relations are either a subset of predefined relations $\mathcal{R}$ or $\{NA\}$ (no relation).

3 Methodology

The overall architecture of LARSON is illustrated in Figure 3. We first extract two kinds of trees and encode plain text using PLMs. With dependency trees and graph attention networks (GATs) (Brody et al., 2021), syntax complemented text representations are derived. On this basis, we extract embeddings of subject entity, object entity, and entity pair-related context (Section 3.1). Then, on the strength of Tree-LSTM (Tai et al., 2015), we lever-

age constituency trees to model subsentences of arbitrary granularity (Section 3.2). Final relations are inferred based on the dynamic fusion of embeddings obtained in section 3.1 and subsentence embeddings obtained in section 3.2. During the process, extra sentence embeddings acquired from both trees are utilized for auxiliary evidence extraction to bring into full play of the syntax information (Section 3.3).

3.1 Text Encoding

Given a document D, we first insert the marker "*" before and after each mention (Zhang et al., 2017). Then, the dependency tree $\mathcal{G}_{dep}$ and constituency tree $\mathcal{G}_{con}$ of each sentence are extracted[2] using Stanza (Qi et al., 2020). We feed tokenized form $\{x_i\}_{i=1}^T$ of D into PLM to obtain contextualized representation $H \in \mathbb{R}^{T \times d}$ and token-level attention matrix[3] $A \in \mathbb{R}^{T \times T}$ where T is the number of tokens and d is the dimension of token embedding:

$$H, A = \text{PLM}([x_1, x_2, ..., x_T]). \quad (1)$$

As pointed out in several studies (Sundararaman et al., 2019; Bai et al., 2021), there is still a big gap between implicitly learned syntax in PLMs and

---

[2]Syntax is parsed in token-level.

[3]Scores of different heads are averaged.

the golden syntax. In LARSON, we alleviate this issue by first incorporating dependency syntax of each sentence which depicts dependencies between words. The contextualized representation H is used as input to the graph attention networks (GATs) (Brody et al., 2021) to encode dependency syntax[4] and aggregate information of neighbor nodes:

$$\mathbf{r}_j = \mathbf{t}^\mathsf{T} \cdot \text{LeakyReLU}\left(W^{(a_1)}H_i + W^{(a_2)}H_j\right),$$

$$\boldsymbol{\alpha} = \text{softmax}([\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{|\mathcal{N}(i)|}]), \quad (2)$$

$$H_i^{(dep)} = \sum_{j \in \mathcal{N}(i)} \boldsymbol{\alpha}_j W^{(a_2)}H_j,$$

where $\mathbf{t} \in \mathbb{R}^{d_1}$, $W^{(a_1)}, W^{(a_2)} \in \mathbb{R}^{d_1 \times d}$ are trainable parameters. Vectors are highlighted in bold. $\mathcal{N}(i)$ contains all neighbor nodes that point to node i according to $\mathcal{G}_{dep}$. After collecting dependency syntax-aware representation $H^{(dep)}$, we can complement the original text representation H by:

$$H^{(c)} = H + H^{(dep)}W^{(z)}, \quad (3)$$

where $W^{(z)} \in \mathbb{R}^{d_1 \times d}$ is a trainable mapping matrix.

Then, representation $\mathbf{e}^{(i)}$ of entity e_i is abstracted by merging the embeddings of markers before each associated mention based on logsumexp (Jia et al., 2019):

$$\mathbf{e}^{(i)} = \log \sum_j \exp(\mathbf{m}^{(i,j)}), \quad (4)$$

where $\mathbf{m}^{(i,j)}$ is the embedding of $m_{(i,j)}$.

In addition to entity-specific embeddings, we also extract the localized context embedding (Zhou et al., 2021) $\mathbf{c}^{(s,o)}$ to represent the entity pair (e_s, e_o):

$$\mathbf{c}^{(s,o)} = \left(H^{(c)}\right)^\mathsf{T} \frac{A_s \odot A_o}{A_s^T A_o}, \quad (5)$$

where $A_s, A_o \in \mathbb{R}^T$ represent subject and object entity's attention score to each token in D, respectively. $\odot$ denotes Hadamard product. Through the

---

[4]Dependency trees are merged by collecting a batch of sentence dependency trees (graphs) and combining them into one graph for efficient graph computation. As a result, each input graph is converted into a disjoint component of the merged graph. During the process, the nodes and edges are relabeled as disjoint segments.

above steps, dependency syntax complemented entity and context representations are acquired. Furthermore, we extract dependency syntax-aware sentence representations by pooling embeddings of all tokens in each sentence using logsumexp (similar to Equation 4) based on $H^{(c)}$.

3.2 Subsentence Modeling

LARSON exploits constituency trees to model subsentences and hierarchical syntax information. Each constituency tree describes a logical way to restore the entire sentence piece by piece. Utilizing constituency trees, we can not only incorporate extra hierarchical syntax information but also encode subsentences of arbitrary granularity. In practice, we first set arbitrary node j in a constituency tree with hidden state $\mathbf{n}^{(j)}$ and memory cell state $\mathbf{d}^{(j)}$. Both $\mathbf{n}^{(j)}$ and $\mathbf{d}^{(j)}$ are initialized with zeros. We set input vectors of all leaf nodes (tokens) with their corresponding representations in the parameter of embedding layer (i.e., embedding matrix E) inside PLM. Input vectors for non-leaf nodes are set to zeros. We then broadcast features of leaf nodes all the way up to the root node by Tree-LSTM (Miwa and Bansal, 2016). Input gate $\mathbf{i}^{(j)}$ and output gate $\mathbf{o}^{(j)}$ of arbitrary node j in $\mathcal{G}_{con}$ are calculated as:

$$\mathbf{i}^{(j)} = \sigma\left(W^{(u)}\mathbf{x}^{(j)} + \sum_{l \in \mathcal{N}(j)} U_l^{(u)}\mathbf{n}^{(j,l)} + \mathbf{b}^{(u)}\right),$$

$$\mathbf{o}^{(j)} = \sigma\left(W^{(v)}\mathbf{x}^{(j)} + \sum_{l \in \mathcal{N}(j)} U_l^{(v)}\mathbf{n}^{(j,l)} + \mathbf{b}^{(v)}\right),$$

$$(6)$$

where σ is sigmoid function. $W, U, \mathbf{b}$ are trainable parameters. $\mathbf{x}^{(j)}$ denotes the input vector of node j. $\mathbf{n}^{(j,l)}$ denotes the hidden state of child l of node j. Integrating input vectors and hidden states is similar to Equation 6, only trainable parameters are newly initialized and sigmoid function is replaced with tanh. Integrated result is marked as $\mathbf{u}^{(j)}$.

Afterwards, we obtain forget gate of child k of node j, i.e., $\mathbf{f}^{(j,k)}$ as:

$$\mathbf{f}^{(j,k)} = \sigma\left(W^{(f)}\mathbf{x}^{(j)} + \sum_{l \in \mathcal{N}(j)} U_{kl}^{(f)}\mathbf{n}^{(j,l)} + \mathbf{b}^{(f)}\right). \quad (7)$$

At last, we update cell state and hidden state:

$$\mathbf{d}^{(j)} = \mathbf{i}^{(j)} \odot \mathbf{u}^{(j)} + \sum_{l \in \mathcal{N}(j)} \mathbf{f}^{(j,l)} \odot \mathbf{d}^{(j,l)},$$

$$\mathbf{n}^{(j)} = \mathbf{o}^{(j)} \odot \tanh(\mathbf{d}^{(j)}), \qquad (8)$$

where $\mathbf{d}^{(j,l)}$ denotes the cell state of child l of node j. Through Equation 6-8, we can extract hidden states of arbitrary nodes in $\mathcal{G}_{con}$ to represent subsentences. To reduce computation complexity, only root nodes of subtrees that have at least two leaf nodes are kept to represent subsentences. Since broadcasting process is tightly related to the structure of tree, each node contains hierarchical syntax information. In order to further complement sentence representations acquired in section 3.1, we adopt GATs of the same architecture as mentioned before (Equation 2) to merge adjacent node features in constituency trees. Then, hierarchical syntax-aware sentence embeddings can be calculated by averaging hidden states of all nodes inside each tree. Later on, we dynamically fuse acquired subsentence and sentence embeddings with various outputs from section 3.1 which will be explained in the next section.

3.3 Dynamic Fusion and Classification

After getting the above representations, we first fuse entity embeddings and context embeddings with subsentence features accordingly using dedicated attention module. Given arbitrary embedding $\mathbf{v} \in \mathbb{R}^d$ and B subsentence embeddings $[\mathbf{n}^{(1)}, \mathbf{n}^{(2)}, ..., \mathbf{n}^{(B)}]$ where $\mathbf{n}^{(i)} \in \mathbb{R}^{d_1}$, the attention score $\boldsymbol{\beta} \in \mathbb{R}^B$ is calculated as (Bahdanau et al., 2014):

$$\mathbf{q}_i = \mathbf{w}^\intercal \cdot \tanh(W^{(b_1)}\mathbf{v} + W^{(b_2)}\mathbf{n}^{(i)}),$$

$$\boldsymbol{\beta} = \mathrm{softmax}([\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_B]), \qquad (9)$$

where $\mathbf{w} \in \mathbb{R}^{d_2}$, $W^{(b_1)} \in \mathbb{R}^{d_2 \times d}$, $W^{(b_2)} \in \mathbb{R}^{d_2 \times d_1}$ are trainable parameters. $\boldsymbol{\beta}$ describes the importance of different subsentences with regarding to different components. We need to emphasize that subsentences of different sentences are all taken into consideration in this step to resolve inter-sentence dependencies. With the guidance of attention score and a trainable mapping matrix $W^{(m)} \in \mathbb{R}^{d \times d_1}$, we can dynamically fuse any embedding $\mathbf{v}$ with subsentence representations using weighted sum:

$$\hat{\mathbf{v}} = \mathbf{v} + W^{(m)} \sum_{i=1}^{B} \boldsymbol{\beta}_i \mathbf{n}^{(i)}. \qquad (10)$$

Statistics	DocRED	CDR	GDA
# Train	3053	500	23353
# Dev	1000	500	5839
# Test	1000	500	1000
# Relations	97	2	2
Avg.# sentences per Doc.	8.0	9.7	10.2

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

Hyperparam	DocRED	CDR	GDA
	BERT	SciBERT	SciBERT
Training epoch	30	30	10
lr for encoder	3e-5	3e-5	2e-5
lr for the rest	2e-4	7e-5	5e-5

Table 2: Hyper-parameters used in three benchmarks.

Through Equation 9-10, we can combine the subject entity embedding, object entity embedding, and context embedding with subsentence representations to form the enhanced representations $\hat{\mathbf{e}}^{(s)}$, $\hat{\mathbf{e}}^{(o)}$, and $\hat{\mathbf{c}}^{(s,o)}$ by replacing $\mathbf{v}$ with $\mathbf{e}^{(s)}$, $\mathbf{e}^{(o)}$, and $\mathbf{c}^{(s,o)}$ respectively. Then, we can calculate the score of relation r (Zhou et al., 2021):

$$\mathbf{z}_s = \tanh(W^{(t_1)}\hat{\mathbf{e}}^{(s)} + W^{(t_2)}\hat{\mathbf{c}}^{(s,o)}),$$

$$\mathbf{z}_o = \tanh(W^{(q_1)}\hat{\mathbf{e}}^{(o)} + W^{(q_2)}\hat{\mathbf{c}}^{(s,o)}), \qquad (11)$$

$$l_{(r|e_s,e_o)} = \mathbf{z}_s^\intercal W^{(r)} \mathbf{z}_o + b^{(r)},$$

where W and b are trainable parameters. In order to reach the full potential of refined syntax, we combine constituency syntax-aware sentence embeddings obtained in section 3.2 with dependency syntax-aware sentence embeddings obtained in section 3.1 using dedicated attention module (Equation 9-10). With the combined sentence embedding $\mathbf{s}^{(i)}$,

we can calculate the probability of sen_i to be an evidence (Xie et al., 2022) as:

$$p_{\left(r_{sen_i}|e_s,e_o\right)} = \sigma\left(\mathbf{s}^{(i)\intercal} W^{(g)} \hat{\mathbf{c}}^{(s,o)} + b^{(g)}\right), \qquad (12)$$

where $W^{(g)}$ and $b^{(g)}$ are trainable parameters.

For relation prediction and evidence extraction of entity pair (e_s, e_o), we adopt adaptive thresholding loss (Zhou et al., 2021) (Equation 13) and bi-

nary cross entropy loss (Equation 14) respectively:

$$\mathcal{L}_{RE} = -\log\left(\frac{\exp(l_{(\text{TH}|e_s,e_o)})}{\sum_{r'\in\mathcal{N}_T\cup\{\text{TH}\}}\exp(l_{(r'|e_s,e_o)})}\right)$$
$$-\sum_{r\in\mathcal{P}_T}\log\left(\frac{\exp(l_{(r|e_s,e_o)})}{\sum_{r'\in\mathcal{P}_T\cup\{\text{TH}\}}\exp(l_{(r'|e_s,e_o)})}\right),$$
$$(13)$$

$$\mathcal{L}_{Evi} = -\sum_{sen_i\in D}[y_i\cdot\log p_{(r_{sen_i}|e_s,e_o)}+$$
$$(1-y_i)\cdot\log(1-p_{(r_{sen_i}|e_s,e_o)})], \quad (14)$$

where $\mathcal{P}_T$ and $\mathcal{N}_T$ denote expressed relations and non-expressed relations respectively. TH is a dummy class introduced to separate positive labels from negative labels. y_i indicates whether sentence sen_i is an evidence.

The overall loss is defined as a combination of $\mathcal{L}_{RE}$ and $\mathcal{L}_{Evi}$ with constant value η:

$$\mathcal{L} = \mathcal{L}_{RE} + \eta\cdot\mathcal{L}_{Evi}. \quad (15)$$

During inference, we follow Xie et al., 2022 and exploit heuristic rules to construct evidence sentences for relation extraction. For more detailed description, we refer interested readers to it.

4 Experiments

4.1 Datasets

In order to fully evaluate our model, we conduct comprehensive experiments on three widely used public DocRE datasets. Statistics of these datasets are listed in Table 1.

- **DocRED** (Yao et al., 2019) is a large-scale crowd-sourced DocRE dataset constructed from Wikipedia articles. It provides 3,053 documents for training covering various domains and requires DocRE models to possess numerous reasoning abilities (e.g., coreference reasoning, or commonsense reasoning).

- **CDR** (Li et al., 2016) is a biomedical DocRE dataset built from 1,500 PubMed abstracts which is randomized into three equal parts for training, validating, and testing. It is manually labeled with binary relations between Chemical and Disease concepts.

- **GDA** (Wu et al., 2019b) is also a biomedical DocRE dataset contains 30,192 MEDLINE

abstracts. The dataset is annotated with binary relations between Gene and Disease concepts using distant supervision. We split GDA according to Christopoulou et al., 2019.

4.2 Implementation Details

LARSON is implemented based on Pytorch (Paszke et al., 2019) and Huggingface's Transformers (Wolf et al., 2020). For all experiments, the number of layers in GATs is set to 3 with only 1 attention head. Output dimensions of GATs are 256. Hidden state and cell state of each node in a constituency tree also share dimension 256. We dropout (Srivastava et al., 2014) attention score inside dedicated attention module with a probability of 50%. Batch size is set to 4. Linear learning rate warmup (Goyal et al., 2017) with ratio 0.06 is deployed followed by a linear decay to 0. η in Equation 15 is set to 0.1. Entire model is optimized by AdamW optimizer (Loshchilov and Hutter, 2017) and tuned on dev set. Mean score of 5 repeated experiments with different random seeds is reported. The rest dataset-specific settings are listed in Table 2.

4.3 Results on DocRED

We compare LARSON with graph-based methods (Zeng et al., 2020; Nan et al., 2020; Wang et al., 2020; Xu et al., 2021b) and PLMs-based methods (Wang et al., 2019; Tang et al., 2020; Ye et al., 2020; Zhou et al., 2021; Zhang et al., 2021; Xie et al., 2022) on DocRED. We report not only F_1 and Ign F_1 (F_1 score excluding the relational facts shared by the training and dev/test set) as the prior studies (Yao et al., 2019), but also Intra F_1 (F_1 that only considers intra-sentence relational facts) and Inter F_1 (F_1 that only considers inter-sentence relational facts). Experimental results listed in Table 3 show that LARSON can achieve leading performance on a general domain DocRE dataset. Specifically, LARSON can improve F_1 score on dev/test set by 0.53%/0.36% over previous state-of-the-art method EIDER (Xie et al., 2022). The advance confirms that syntax information and subsentence modeling are two crucial factors in DocRE. More specifically, we can observe performance boosts in both Inter F_1 (+0.54%) and Intra F_1 (+0.16%), indicating the extensiveness of our method and the efficiency in targeting inter-sentence instances.

Model	Dev				Test	
	Ign F_1	F_1	Intra F_1	Inter F_1	Ign F_1	F_1
Graph-based Methods						
LSR-BERT$_{base}$ (Nan et al., 2020)	52.43	59.00	65.26	52.05	56.97	59.05
GLRE-BERT$_{base}$ (Wang et al., 2020)	-	-	-	-	55.40	57.40
HeterGSAN-BERT$_{base}$ (Xu et al., 2021b)	58.13	60.18	-	-	57.12	59.45
GAIN-BERT$_{base}$ (Zeng et al., 2020)	59.14	61.22	67.10	53.90	59.00	61.24
PLMs-based Methods						
BERT$_{base}$ (Wang et al., 2019)	-	54.16	61.61	47.15	-	53.20
BERT-TS$_{base}$ (Wang et al., 2019)	-	54.42	61.80	47.28	-	53.92
HIN-BERT$_{base}$ (Tang et al., 2020)	54.29	56.31	-	-	53.70	55.60
CorefBERT$_{base}$ (Ye et al., 2020)	55.32	57.51	-	-	54.54	56.96
ATLOP-BERT$_{base}$ (Zhou et al., 2021)	59.22	61.09	-	-	59.31	61.30
DocuNet-BERT$_{base}$ (Zhang et al., 2021)	59.86	61.83	-	-	59.93	61.86
EIDER-BERT$_{base}$ (Xie et al., 2022)	60.51	62.48	68.47	55.21	60.42	62.47
LARSON-BERT$_{base}$	**61.05**	**63.01**	**68.63**	**55.75**	**60.71**	**62.83**

Table 3: Results (%) of relation extraction on the dev and test set of DocRED. The best result on test set is reported according to submissions on CodaLab. Results of other methods are directly taken from original papers.

Model	CDR	GDA
LSR-BERT (Nan et al., 2020)	64.8	82.2
SciBERT (Zhou et al., 2021)	65.1	82.5
DHG-BERT (Zhang et al., 2020)	65.9	83.1
GLRE-SciBERT (Wang et al., 2020)	68.5	-
ATLOP-SciBERT (Zhou et al., 2021)	69.4	83.9
EIDER-SciBERT (Xie et al., 2022)	70.63	84.54
LARSON-SciBERT	**71.59**	**86.02**

Table 4: Results (%) of relation extraction on test set of CDR and GDA. We choose the best checkpoint based on dev set to evaluate the final performance. Result of SciBERT is based on the re-implemented version (Zhou et al., 2021). Other results are directly taken from original papers.

Ablation	Ign F_1	F_1	Intra F_1	Inter F_1
LARSON-BERT	**61.05**	**63.01**	68.63	**55.75**
w/o dependency tree	60.77	62.73	68.62	55.22
w/o constituency tree	60.54	62.60	**68.69**	55.02
w/o dynamic fusion	60.24	62.46	68.36	55.20

Table 5: Ablation study of LARSON on dev set of DocRED.

4.4 Results on biomedical Datasets

Besides general domain DocRE dataset DocRED, we also compare LARSON with various advanced methods (Nan et al., 2020; Zhang et al., 2020; Wang et al., 2020; Zhou et al., 2021; Xie et al., 2022) on two biomedical domain datasets CDR and GDA. Experimental results are listed in Table 4. In summary, LARSON achieves significant improvements over two tested datasets (+0.96% F_1

on CDR and +1.48% F_1 on GDA). As we can see in Table 1, CDR and GDA have more sentences in a document on average compared to DocRED. In these complex documents, LARSON manages to skillfully conduct relation extraction, proving the reasonableness and capability of designed architecture. The fact that LARSON can work well in biomedical domain further demonstrates its generality.

4.5 Ablation Study

To exhaustively understand how each component contributes to final performance, we conduct three ablation studies and list the results in Table 5. *w/o dependency tree* removes dependency syntax encoding module. All output embeddings in section 3.1 are calculated based on plain output of PLM. We can observe that without dependency syntax, decline in Inter F_1 (-0.53%) is much more obvious than it in Intra F_1 (-0.01%). Similar trends happen when we remove constituency syntax and subsentence modeling as in the *w/o constituency tree*. Inter F_1 decreases to 55.02% which is even worse than *w/o dependency tree*. Surprisingly consistent results prove that both dependency and constituency syntax are crucial when it comes to inter-sentence instances. Noting that while the overall performance drops in *w/o constituency tree*, Intra F_1 uncommonly increases. It suggests there is still room for improving how constituency information is integrated, especially in the intra-sentence scenario. *w/o dynamic fusion* removes dynamic fusing

of output embeddings in section 3.1 and section 3.2. Instead, we calculate the average of different subsentence embeddings and directly add them to entity/context embeddings after being mapped to proper shapes. Dependency syntax-aware and constituency syntax-aware sentence embeddings are directly added together in a sentence-wise manner to carry out auxiliary task. Without dynamic fusion, the performance of LARSON reduces dramatically in all aspects which fully demonstrates the necessity of this module. Different subsentences are associated with different entities. Brutally combining them together using equal weights neglects the interconnections between the two components.

4.6 Case Study

To intuitively demonstrate that LARSON can link entity pairs with instructive subsentences, we take one example in dev set of DocRED and visualize it in Figure 4. Twenty subsentences with the highest attention scores are colored pink in different depths. Similar ones like *Proclamation* and *Emancipation Proclamation* are merged. As shown in Figure 4, LARSON assigns extremely high relevance score to event *Emancipation Proclamation*, role *freed slaves*, place *Route 28*, role *land developer*, and other characters *his subdivisions*. Among which, human can infer the relation between *William* and *United States* by a key subsentence "*African-American land developer*" in the 3rd sentence. The fact that LARSON also views *land developer* as a highly relevant subsentence indicates our model can efficiently capture decisive details. Besides that, all marked subsentences have direct or indirect connections with at least one of subject and object. This example is a strong proof that with subsentence modeling, LARSON can filter out inconsequential pieces and better concentrate on meaningful information.

5 Related Work

Relation extraction (RE) is an indispensable part of many real-world applications like question answering (Hixon et al., 2015), biomedical text analysis (Hong et al., 2020), etc. Many studies have been done to solve sentence-level RE where entities are within the same sentence, such as sequence-based methods (Zeng et al., 2014; Zhang et al., 2017), graph-based methods (Miwa and Bansal, 2016; Wu et al., 2019a), and PLMs based methods (Han et al., 2021; Zhou and Chen, 2021). Nguyen
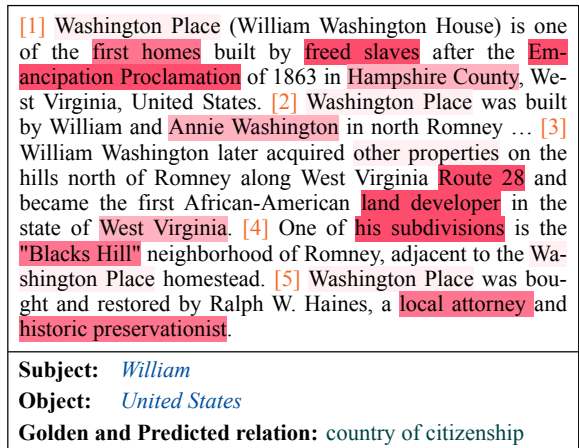

Figure 4: Attention weights of an example in DocRED with regarding to entity pair (*William*, *United States*).

and Moschitti, 2011 even successfully utilized both dependency and constituency trees to better solve sentence-level RE task. These methods have shown tremendous performance in sentence-level RE task. Nevertheless, according to Yao et al., 2019, a considerable proportion of relation instances can only be inferred from multiple sentences, and existing methods are still inadequate for real-world RE problems. DocRE, due to its more realistic setting, has gained more and more attention lately. With the rapid development of PLMs, they have been widely adopted as the first step in DocRE to encode plain text. According to the rest steps, existing high performance methods can be divided into two categories. One category inherits implicitly learned attention distribution inside PLMs or designs extra attention modules to capture long range dependencies. Yuan et al., 2021 uses PLMs to encode semantic features of document and weights different sentences based on different entities via attention mechanism. In order to combine sentence-level and document-level features together, gating mechanism is developed. Instead, Zhou et al., 2021 inherits implicitly learned attention distribution inside PLMs to determine entity-related context. In addition, they also adopt adaptive threshold to better distinguish positive labels from negative labels. The other category exploits GCNs to capture complex interactions between different components (e.g., entities, or sentences) and conduct logical reasoning. Zeng et al., 2020 proposes a heterogeneous mention-level graph to capture complex interactions among different mentions. They further aggregate different mention representations to construct an entity-level graph. Based on the graphs,

they develop a novel path reasoning mechanism for final relation extraction. Xu et al., 2021b encourages the model to reconstruct reasoning paths while identifying correct relations. Unlike previous works, LARSON tries to better solve DocRE task by integrating explicitly refined syntax and subsentence modeling. To the best of our knowledge, LARSON is the first work investigating the effects of subsentences under the DocRE scenario.

6 Conclusion

In this work, we propose a novel LARSON model for document-level relation extraction task. LARSON mostly exploits two kinds of extra syntax information, namely dependency syntax and constituency syntax. Graph attention networks and Tree-LSTM are adopted to encode the two kinds of information. Furthermore, through leveraging dedicated attention module, we can dynamically weight different subsentences to assist LARSON in capturing instructive information regardless of the granularity. Experiments on three public DocRE datasets demonstrate that our LARSON model outperforms existing methods by a large margin.

Limitations

The challenge to extract instructive information not only exists in DocRE but also in many other document-level tasks (e.g., reading comprehension, document retrieval). For now, our hypothesis is merely tested in DocRE.

Acknowledgements

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

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.

Shaked Brody, Uri Alon, and Eran Yahav. 2021. How attentive are graph attention networks? *arXiv preprint arXiv:2105.14491*.

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.

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.

Zhichao Duan, Xiuxing Li, Zhengyan Zhang, Zhenyu Li, Ning Liu, and Jianyong Wang. 2021. Bridging the language gap: Knowledge injected multilingual question answering. In *2021 IEEE International Conference on Big Knowledge (ICBK)*, pages 339–346. IEEE.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv: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.

Xu Han, Weilin Zhao, Ning Ding, Zhiyuan Liu, and Maosong Sun. 2021. Ptr: Prompt tuning with rules for text classification. *arXiv preprint arXiv:2105.11259*.

Ben Hixon, Peter Clark, and Hannaneh Hajishirzi. 2015. Learning knowledge graphs for question answering through conversational dialog. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 851–861, Denver, Colorado. Association for Computational Linguistics.

Lixiang Hong, Jinjian Lin, Shuya Li, Fangping Wan, Hui Yang, Tao Jiang, Dan Zhao, and Jianyang Zeng. 2020. A novel machine learning framework for automated biomedical relation extraction from large-scale

literature repositories. *Nature Machine Intelligence*, 2(6):347–355.

Robin Jia, Cliff Wong, and Hoifung Poon. 2019. Document-level n-ary relation extraction with multi-scale 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.

Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

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.

Xiuxing Li, Zhenyu Li, Zhengyan Zhang, Ning Liu, Haitao Yuan, Wei Zhang, Zhiyuan Liu, and Jianyong Wang. 2022. Effective few-shot named entity linking by meta-learning. In *2022 IEEE 38th International Conference on Data Engineering (ICDE)*, pages 178–191. IEEE Computer Society.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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.

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.

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.

Truc-Vien T. Nguyen and Alessandro Moschitti. 2011. Joint distant and direct supervision for relation extraction. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 732–740, Chiang Mai, Thailand. Asian Federation of Natural Language Processing.

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*, pages 101–108, Online. Association for Computational Linguistics.

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.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958.

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*.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. 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 1556–1566, Beijing, China. Association for Computational Linguistics.

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. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 197–209. Springer.

Difeng Wang, Wei Hu, Ermei Cao, and Weijian Sun. 2020. Global-to-local neural networks for document-level relation extraction. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3711–3721, Online. Association for Computational Linguistics.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi 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 Rush. 2020. Transformers: State-of-the-art natural language processing.

In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Felix Wu, Amauri Souza, Tianyi Zhang, Christopher Fifty, Tao Yu, and Kilian Weinberger. 2019a. Simplifying graph convolutional networks. In *International conference on machine learning*, pages 6861–6871. PMLR.

Ye Wu, Ruibang Luo, Henry Leung, Hing-Fung Ting, and Tak-Wah Lam. 2019b. Renet: A deep learning approach for extracting gene-disease associations from literature. In *International Conference on Research in Computational Molecular Biology*, 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, Dublin, Ireland. Association for Computational Linguistics.

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 *The 35th AAAI Conference on Artificial Intelligence (AAAI-21)*.

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.

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. *arXiv preprint arXiv:2004.06870*.

Changsen Yuan, Heyan Huang, Chong Feng, Ge Shi, and Xiaochi Wei. 2021. Document-level relation extraction with entity-selection attention. *Information Sciences*, 568:163–174.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344, Dublin, Ireland. Dublin City University and 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.

Ningyu Zhang, Xiang Chen, Xin Xie, Shumin Deng, Chuanqi Tan, Mosha Chen, Fei Huang, Luo Si, and Huajun Chen. 2021. Document-level relation extraction as semantic segmentation. *arXiv preprint arXiv:2106.03618*.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. 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.

Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Tingwen Liu, Hengzhu Tang, Wang Yubin, and Li Guo. 2020. Document-level relation extraction with dual-tier heterogeneous graph. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1630–1641, Barcelona, Spain (Online). International Committee on 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 preprint arXiv:2201.04826*.

Wenxuan Zhou and Muhao Chen. 2021. An improved baseline for sentence-level relation extraction. *arXiv preprint arXiv:2102.01373*.

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.