Document-level Relation Extraction with Dual-tier Heterogeneous Graph

Zhenyu Zhang, Bowen Yu, Xiaobo Shu, Tingwen Liu[*]
Hengzhu Tang, Yubin Wang and **Li Guo**
Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{zhangzhenyu1996,yubowen,shuxiaobo,liutingwen}@iie.ac.cn
{tanghengzhu,wangyubin,guoli}@iie.ac.cn

Abstract

Document-level relation extraction (RE) poses new challenges over its sentence-level counterpart since it requires an adequate comprehension of the whole document and the multi-hop reasoning ability across multiple sentences to reach the final result. In this paper, we propose a novel graph-based model with Dual-tier Heterogeneous Graph (DHG) for document-level RE. In particular, DHG is composed of a structure modeling layer followed by a relation reasoning layer. The major advantage is that it is capable of not only capturing both the sequential and structural information of documents but also mixing them together to benefit for multi-hop reasoning and final decision-making. Furthermore, we employ Graph Neural Networks (GNNs) based message propagation strategy to accumulate information on DHG. Experimental results demonstrate that the proposed method achieves state-of-the-art performance on two widely used datasets, and further analyses suggest that all the modules in our model are indispensable for document-level RE.

1 Introduction

Relation extraction (RE), which aims to identify relational facts between entities from plain text, is one of the most fundamental tasks in information extraction (IE) and natural language processing (NLP). Existing methods usually focus on extracting relations from a single sentence (i.e., sentence-level RE) (Soares et al., 2019; Yu et al., 2019; Zhang et al., 2020a). However, sentence-level RE suffers from a non-ignorable limitation in practice: a large number of relational facts are expressed in multiple sentences. Lately, the attention of identifying inter-sentence relations heightens the interest of moving RE forward from sentence-level to document-level.

The most straightforward way to perform document-level RE is to treat documents as long sequences and then employ sequential models, adapted from sentence-level RE, to extract the relation between given entities (Gu et al., 2017; Li et al., 2018a; Verga et al., 2018). However, these methods inevitably face several challenges in modeling long-term dependencies and enabling multi-hop reasoning, yet identifying long-term and inter-sentence relations is precisely the key and difficult point of document-level RE. Recently, a number of exquisite graph-based models for document-level NLP tasks show their sparkles in capturing multi-hop relations (De Cao et al., 2019; Tu et al., 2019; Christopoulou et al., 2019). Nevertheless, these approaches often fail to adequately capture the inherent structure of documents and discard masses of valuable structural information when transforming documents into graphs. As a matter of fact, the document structure, especially the positional, syntactical, and hierarchical structure, has proven to be very effective for many document-level tasks (Miculicich et al., 2018; Li et al., 2018b).

Look at a concrete example shown in Figure 1, in order to predict the relation between *bradycardia* and *ramipril*, one has to first grasp the key point that *bradycardia* is caused by *hyperpotassemia* in sentence 1, then identify the fact that the cause of *hyperkalemia* is *ramipril* from sentence 3, and finally infer from these observations that *ramipril* is also the cause of *bradycardia* with a background coreference knowledge that *hyperpotassemia* and *hyperkalemia* refer to the exact same entity. It demonstrates that the decision-making process in document-level RE requires understanding first and reasoning later over

[*] Corresponding author.

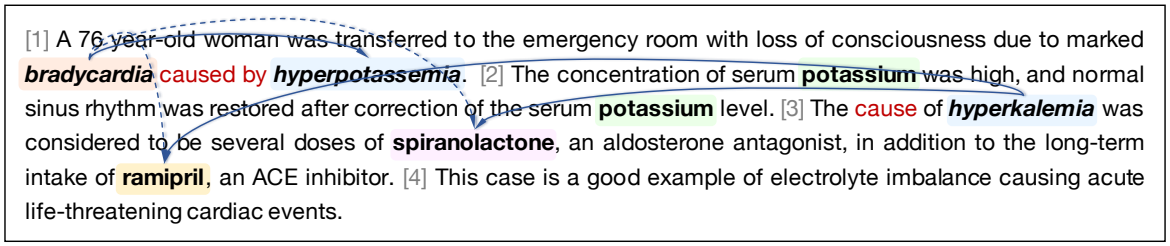

Figure 1: An example adapted from the CDR dataset, in which the document is annotated with named entity mentions (words with bold font), coreference information (mentions with the same background color), intra-sentence and inter-sentence relations (solid and dotted lines).

multiple sentences, which is obviously beyond the reach of traditional sequential approaches. Furthermore, there are amounts of useful structural information to help detect the relation between *bradycardia* and *ramipril*, such as the keyword *caused by* is close to *bradycardia* in the first sentence and these two mentions of *hyperkalemia* belong to two different sentences, which provides an in-depth understanding of the whole text and should be fully exploited in the document modeling process.

In this paper, we propose a novel document-level RE model that builds a Dual-tier Heterogeneous Graph (DHG) to successively model document structure and enable relational reasoning. Specifically, the first-tier of DHG is a structure modeling layer (SML) that responsible for comprehensively encoding the inherent structure of document from three aspects of sequence, syntax, and hierarchy, thus each document is transformed into a graph in which nodes are words and sentences whereas edges are relationships between them. In the second-tier, based on the semantically-rich representations induced from the previous layer, a relation reasoning layer (RRL) is introduced to propagate relational information among various entities and enable multi-hop relational reasoning. With Graph Neural Networks (GNNs), we assume that the desired signal for identifying relational facts could be captured by propagating node information along edges in our DHG. By this means, taking Figure 1 as an example again, the messages of keywords *caused by* and *cause* will be propagated to word nodes through sentence nodes in SML, and gradually accumulated to the entity nodes in RRL for the final decision. To the best of our knowledge, it is the first attempt to separate document modeling from multi-hop reasoning in document-level NLP tasks via a dual-tier heterogeneous structure.

We conduct extensive experiments on two public widely used document-level RE datasets. Results suggest that the proposed model achieves state-of-the-art performance. Through detailed ablation studies, we further show that all the components in our approach are indispensable for document-level RE. Moreover, we also demonstrate that incorporating pre-trained language model (e.g., BERT (Devlin et al., 2019)) with DHG can bring further improvements.

2 Preliminaries

2.1 Problem Statement

First of all, we define the document-level RE task in a formal way. Given an annotated document $\mathcal{D} = \{\mathcal{S}_i\}_{i=1}^{n_s}$ and its entity set $\mathcal{V} = \{\mathcal{E}_i\}_{i=1}^{n_e}$, where $\mathcal{S}_i = \{w_j\}_{j=1}^{n_w^i}$ denotes the i-th sentence with n_w^i words and $\mathcal{E}_i = \{m_j\}_{j=1}^{n_m^i}$ is the i-th entity with n_m^i entity mentions. The ultimate goal is to predict all intra- and inter-sentence relations $\mathcal{R}' \in \mathcal{R} = \{r_i\}_{i=1}^{n_r}$ between each entity pair. For simplification, in the remainder of this paper, we ignore the superscript i that indicates the element number of the i-th sentence or entity. Note that many relational facts are expressed in multiple sentences, which means the document-level RE task is more difficult than traditional sentence-level task, and the document-level RE model should have a strong ability in semantic modeling and relational reasoning.

2.2 Message Propagation Strategy

We now define how information propagates over DHG. Typically, graph-based models follow a layer-wise propagation manner that all the nodes update simultaneously in each layer, and different variants

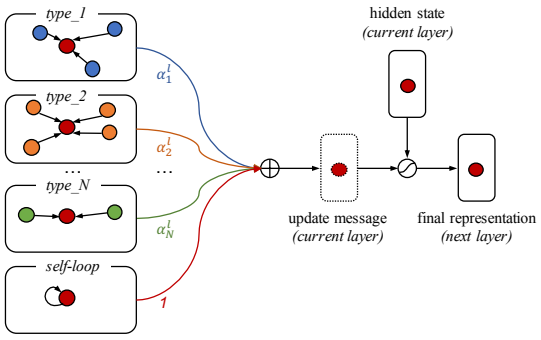

Figure 2: A toy illustration of the message propagation process in the l-th layer of WR-GCN.

of GNNs have different implementations of message propagation strategies. In this study, we follow the basic propagation idea of Relational Graph Convolutional Networks (R-GCN) (Schlichtkrull et al., 2018), which can handle high-relational data characteristics and make full use of different edge types. Formally, at l-th layer, given the hidden state $\mathbf{h}_i^l \in \mathbb{R}^d$ of node i and its neighbors $\mathcal{N}_i$ with corresponding edge types $\mathcal{T}$, R-GCN propagates message across different neighboring nodes and generates transformed representation in the next layer for node i via

$$\mathbf{h}_i^{l+1} = \sigma(\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}_i^t} \frac{1}{|\mathcal{N}_i^t|} \mathbf{W}_t^l \mathbf{h}_j^l + \mathbf{W}_s^l \mathbf{h}_i^l), \tag{1}$$

where $\mathbf{W}_t^l \in \mathbb{R}^{d \times d}$ refers an edge type-specific weight matrix, $\mathbf{W}_s^l \in \mathbb{R}^{d \times d}$ is a general matrix for self-connection and d is the dimension of hidden states. Obviously, different types of edges in heterogeneous graph usually have different importance and should be treated differential. To model such a diversity, we propose to assign unequal weights to different edge types and modify R-GCN in Equation 1 to

$$\mathbf{h}_i^{l+1} = \sigma(\sum_{t \in \mathcal{T}} \sum_{j \in \mathcal{N}_i^t} \frac{\alpha_t^l}{|\mathcal{N}_i^t|} \mathbf{W}_t^l \mathbf{h}_j^l + \mathbf{W}_s^l \mathbf{h}_i^l), \tag{2}$$

where α_t^l is a trainable parameter to model the interaction strength between two adjacent nodes with type t in the l-th layer.

Furthermore, it has been shown that GNNs usually suffer from the over-smoothing problem if the number of layers is large (Kipf and Welling, 2017), making different nodes have similar representations and lose the distinction among nodes. To tackle this problem, we add a gating mechanism (Gilmer et al., 2017) to control the extent of propagating the update message to the next layer, in which the update message $\mathbf{u}_i^l$ can be obtained via Equation 2 without non-linear activation function σ. The gate-level is computed by $\mathbf{u}_i^l$ and $\mathbf{h}_i^l$ with a linear transformation $\mathcal{F}_g$, and the final representation is defined as a gated combination of previous features and a non-linear transformation of update message:

$$g_i^l = \text{sigmoid}(\mathcal{F}_g([\mathbf{u}_i^l; \mathbf{h}_i^l])), \tag{3}$$

$$\mathbf{h}_i^{l+1} = g_i^l \odot \tanh(\mathbf{u}_i^l) + (1 - g_i^l) \odot \mathbf{h}_i^l, \tag{4}$$

where $\odot$ stands for element-wise multiplication. For brevity, we abbreviate the message propagation process as WR-GCN (Weighted Relational Graph Neural Networks) in the remainder of this paper. Figure 2 shows the workflow of one layer with WR-GCN.

3 Methodology

In this section, we introduce the proposed dual-tier heterogeneous graph (DHG) and document-level RE model. Figure 3 shows the overall system diagram. Specifically, the RE model can be categorized into

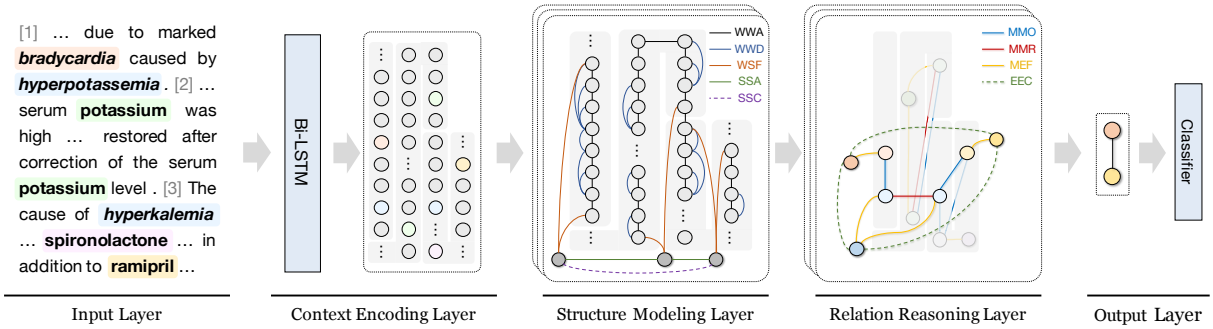

Figure 3: The abstract architecture of our proposed model. The dual-tier heterogeneous graph aims to update word node representations and aggregate messages into final representations of entity nodes. There are two kinds of nodes in the structure modeling layer (SML) and relation reasoning layer (RRL) respectively. In SML, we utilize light gray nodes to denote words and dark gray to sentences. In RRL, nodes with plane style refer to mentions and shaded nodes indicate entities. For a good visualization, we only exhibit some representative edges and nodes in this figure.

five layers: (1) *Input Layer* is responsible for transforming input words into dense vectorized representations; (2) *Context Encoding Layer* could be any common sequence encoder to generate a contextualized representation for each word; (3) *Structure Modeling Layer* is the first-tier heterogeneous graph of DHG, aiming to model the inherent structural information of plain text, including adjacency, affiliation, and syntactic dependency relations; (4) *Relation Reasoning Layer* focuses on capturing multi-hop relations between entity pairs in a document, which is the second-tier heterogeneous graph of DHG; (5) *Output Layer* treats relation prediction as a multi-label classification problem and predicts possible relations for each entity pair.

3.1 Input Layer

The input layer embeds both semantic and augmented information of words into their input features. To be more specific, we use d_w-dimensional word embedding as basic features to capture meaningful semantic regularities. Meanwhile, two extra features are also used to augment the input. The first one is type embedding, which is used to embed the entity type for each mention word and has been proved to be very useful for RE in previous work (Zhang et al., 2018; Christopoulou et al., 2019). The second one is coreference embedding, which is used to mark which entity the word belongs to and help the model catch non-local coreference information. Finally, for each word w_i, we concatenate its *word embedding* $\mathbf{w}_i$, *type embedding* $\mathbf{t}_i$ and *coreference embedding* $\mathbf{c}_i$ to build input features $\mathbf{x}_i = [\mathbf{w}_i; \mathbf{t}_i; \mathbf{c}_i] \in \mathbb{R}^{d_x}$, where $[\cdot;\cdot]$ denotes concatenation operator and $d_x = d_w + d_t + d_c$.

3.2 Context Encoding Layer

We regard the whole document as a long sequence with n words, then a Bi-LSTM network is adopted to encode the contextual information for each word[1]. For simplicity, we denote the operation of an LSTM unit on $\mathbf{x}_i$ as $\text{LSTM}(\mathbf{x}_i)$, the contextualized word representation can be obtained as

$$\mathbf{h}_i = \mathcal{F}([\overrightarrow{\text{LSTM}}(\mathbf{x}_i); \overleftarrow{\text{LSTM}}(\mathbf{x}_i)]), \tag{5}$$

where $\mathbf{h}_i \in \mathbb{R}^{d_h}$ and $\mathcal{F} : \mathbb{R}^{2\times d_h} \rightarrow \mathbb{R}^{d_h}$ refers to a linear function, in which d_h indicates the hidden size of LSTM units. In this way, we can efficiently capture the past (via forward states) and future (via backward states) features for a specific time. As a result, we use $\mathbf{H}_W = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_n\}$ to denote all word representations generated for input sequence.

[1]Theoretically, it could be any sequence encoder, including ELMo (Peters et al., 2018), BERT (Devlin et al., 2019) or other advanced architectures. Here we take Bi-LSTM as an example.

3.3 Structure Modeling Layer

In the first layer of DHG, we treat each *word* and *sentence* in the document as a node. It is intuitive, because a document is made up of many sentences, and a sentence is made up of many words. Naturally, we can model the inherent structure of a document with the following five types of edges:

- **w**ord-**w**ord **a**djacency (WWA): To keep the word-level sequential structure, an edge is established between two word nodes if they are adjacent in the document.
- **w**ord-**w**ord **d**ependency (WWD): To encode the syntactical structure, two word nodes are connected if they are neighboring in the sentence-level dependency tree.
- **w**ord-**s**entence a**ff**iliation (WSF): To model the hierarchical structure of documents, we connect a word node and a sentence node if the word resides in the sentence.
- **s**entence-**s**entence **a**djacency (SSA): To maintain the sentence-level sequential structure, an edge between two sentence nodes is built if they are adjacent in the document.
- **s**entence-**s**entence **c**omplement (SSC): To enhance the connectivity of graph, all sentence node pairs that do not meet the SSA condition are connected.

In the structure modeling layer (SML), we parse dependency tree for each sentence separately and directly utilize the outputs of context encoding layer as initial features of word nodes. A max-pooling operation is applied over all word nodes in a sentence to obtain the sentence node representation: $\mathbf{s} = \max\{\mathbf{h}_j\}_{j=1}^{n_w}$. Afterward, the message passing strategy introduced in Section 2.2 is used to update the representations of word and sentence nodes:

$$(\bar{\mathbf{H}}_W, \bar{\mathbf{H}}_S) = \text{WR-GCN}_{\text{SML}}(\mathbf{H}_W, \mathbf{H}_S), \tag{6}$$

where $\mathbf{H}_S = \{\mathbf{s}_1, \ldots, \mathbf{s}_{N_s}\}$ is the set of sentence node representations. For each word node, we concatenate its features before and after WR-GCN$_{\text{SML}}$ as its output representation: $\hat{\mathbf{h}}_i = \mathcal{F}([\mathbf{h}_i; \bar{\mathbf{h}}_i])$. Such a shortcut connection mechanism is able to combine both sequential and structural features, and provides a solid foundation for the next reasoning step.

3.4 Relation Reasoning Layer

The second tier of DHG is constructed for graph-based reasoning, which is expected to first exploit and propagate relational information among entity mentions, and then summarize them into corresponding entities. Inspired by the success of Entity-GCN (De Cao et al., 2019), we treat *mention* and *entity* as nodes and design the following four types of edges:

- **m**ention-**m**ention co**o**ccurrence (MMO): To pledge the performance of intra-sentence relation, two mention nodes are connected if they reside in the same sentence.
- **m**ention-**m**ention co**r**eference (MMR): To capture non-local relationship among mentions, an edge between two mention nodes is built if they refer to the same entity.
- **m**ention-**e**ntity a**ff**iliation (MEF): To pass the mention-level message to entity-level, we connect a mention node and an entity node if the mention refers to the entity.
- **e**ntity-**e**ntity **c**omplement (EEC): To prevent having disconnected graphs and enhance the multi-hop reasoning ability, all entity nodes are connected with each other.

In this layer, for an entity mention m ranging from the s-th word to the t-th word in text, we initialize its representation as $\mathbf{m} = \frac{1}{s-t+1}\sum_{k=s}^{t} \hat{\mathbf{h}}_k$, and the representation of an entity e is computed as the average of all its mention features: $\mathbf{e} = \frac{1}{n_m}\sum_j \mathbf{m}_j$. Similar to SML, WR-GCN is also employed to propagate messages among nodes:

$$(\bar{\mathbf{H}}_E, \bar{\mathbf{H}}_M) = \text{WR-GCN}_{\text{RRL}}(\mathbf{H}_E, \mathbf{H}_M), \tag{7}$$

where $\mathbf{H}_M$ and $\mathbf{H}_E$ are representation sets of mention and entity nodes respectively. After L times message passing, all nodes will have their final representations.

3.5 Output Layer

To determine the semantic relations between two entities, we treat the relation prediction as a multi-label classification problem. In particular, for each entity pair (e_i, e_j), we concatenate there entity features with *relative distance embeddings*, and use a bilinear function to calculate the probability for each relation:

$$\hat{\mathbf{e}}_i = [\bar{\mathbf{e}}_i; \mathbf{d}_{ij}], \quad \hat{\mathbf{e}}_j = [\bar{\mathbf{e}}_j; \mathbf{d}_{ji}], \tag{8}$$

$$\mathbf{y} = \text{sigmoid}(\hat{\mathbf{e}}_i^\top \mathbf{W} \hat{\mathbf{e}}_j + b), \tag{9}$$

where $\mathbf{d}_{ij} \in \mathbb{R}^{d_d}$ and $\mathbf{d}_{ji} \in \mathbb{R}^{d_d}$ are relative distance embeddings between the first mentions of two entities in the document, $\mathbf{W} \in \mathbb{R}^{d \times n_r \times d}$ is a learned bi-affine tensor, $b \in \mathbb{R}^d$ is the bias vector (in which $d = d_h + d_d$) and $\mathbf{y} \in \mathbb{R}^{n_r}$ denotes the prediction for all relations. Finally, the loss function is defined as the sum of binary cross-entropy between gold annotation and its predicted probability for each fact.

4 Experiments

4.1 Datasets

We evaluate the proposed model on two public document-level RE datasets: (1) **CDR** (BioCreative V): The Chemical-Disease Reactions dataset created by Li et al. (2016) is the most widely used dataset for document-level RE, which is manually annotated with binary interactions between Chemical and Disease concepts. There are 1,500 PubMed abstracts with 3,116 relational facts in the dataset, and it is split into three equal-sized sets for training, development, and test. (2) **GDA** (DisGeNet): The Gene-Disease Associations dataset is a recent document-level RE dataset released by Wu et al. (2019), it contains 30,192 MEDLINE abstracts and 46,343 relational facts altogether, in which 29,192 abstracts are used for training and 1,000 for test. The dataset is annotated with binary interactions between Gene and Disease concepts at document-level with distant supervision hypothesis. Following Christopoulou et al. (2019), we split the original training set into an 80/20 percentage split as training and development sets.

4.2 Compared Models

We compare the proposed model against the following baseline models for document-level RE: (1) **CD-REST** (Xu et al., 2016): It is an end-to-end model with support vector machines and several manual features. (2) **Syn-Sem** (Zhou et al., 2016): It consists of a feature-based model, a kernel-based model, and a neural network model to fully utilize lexical, syntactic, and semantic information. (3) **ME-CNN** (Gu et al., 2017): It combines a maximum entropy model and a convolutional neural network model to extract both inter- and intra-sentence relations. (4) **RPCNN** (Li et al., 2018a): It proposes a document-level recurrent piecewise convolutional neural network with attention, piecewise pooling, and multi-instance learning strategies. (5) **BRAN** (Verga et al., 2018): It presents a bi-affine relation attention network with a self-attention encoder, which simultaneously scores all mention pairs within a document. (6) **C-CHAR** (Nguyen and Verspoor, 2018): It incorporates character-based word representations into CNN-based model, and outperforms several CNN/RNN/Attention-based models. (7) **GCNN** (Sahu et al., 2019): It builds a labeled edge graph convolutional neural network over a document-level graph, which is the first attempt to use GNNs in document-level RE. (8) **EoG** (Christopoulou et al., 2019): It constructs an edge-oriented graph and uses an iterative algorithm over the graph edges, which is the recent state-of-the-art on the CDR dataset, and **Full** is a variant of EoG that uses a fully connected graph as inputs. We denote our model as **DHG** and implement two versions **DHG-LSTM** and **DHG-BERT** using BiLSTM and BERT in the *Context Encoding Layer*, respectively. Meanwhile, we also remove the DHG structure to establish the baseline model **LSTM-DRE** and **BERT-DRE** for direct comparisons.

4.3 Implementation Details

Following popular choices and previous work, we employ the 200-dimensions PubMed pre-trained word embeddings (Chiu et al., 2016) for the CDR dataset and random word embeddings drawn from a uniform distribution $[-0.05, 0.05]$ for the GDA dataset. Mentions that are not grounded to a Knowledge Base ID

Data	Model	Overall (%)			Intra (%)			Inter (%)		
		P	R	F_1	P	R	F_1	P	R	F_1
CDR	CD-REST (Xu et al., 2016)	59.6	44.0	50.7	-	-	-	-	-	-
	Syn-Sem (Zhou et al., 2016)	64.8	49.2	56.0	-	-	-	-	-	-
	RPCNN (Li et al., 2018a)	55.2	63.6	59.1	-	-	-	-	-	-
	ME-CNN (Gu et al., 2017)	55.7	68.1	61.3	59.7	55.0	57.2	51.9	7.0	11.7
	BRAN (Verga et al., 2018)	55.6	70.8	62.1	-	-	-	-	-	-
	C-CHAR (Nguyen and Verspoor, 2018)	57.0	68.6	62.3	-	-	-	-	-	-
	GCNN (Sahu et al., 2019)	52.8	66.0	58.6	-	-	-	-	-	-
	Full (Christopoulou et al., 2019)	59.1	56.2	57.6	71.2	62.3	66.5	37.1	42.0	39.4
	EoG (Christopoulou et al., 2019)	62.1	65.2	63.6	64.0	73.0	68.2	56.0	46.7	50.9
	LSTM-DRE	53.1	64.3	58.2	61.7	69.3	65.3	43.5	45.9	44.7
	DHG-LSTM	61.2	68.7	**64.7**	64.4	73.3	**68.6**	51.5	56.9	**54.1**
	BERT-DRE	54.2	67.7	60.2	63.6	72.9	68.0	43.4	47.5	45.4
	DHG-BERT	61.8	70.5	**65.9**	65.3	75.8	**70.1**	51.2	58.6	**54.6**
GDA	Full (Christopoulou et al., 2019)†	-	-	79.9	-	-	84.6	-	-	**54.8**
	EoG (Christopoulou et al., 2019)†	-	-	80.2	-	-	84.7	-	-	45.7
	LSTM-DRE	77.4	79.3	78.4	82.3	83.7	83.0	42.9	46.0	44.3
	DHG-LSTM	80.3	84.4	**82.2**	84.0	86.8	**85.4**	50.5	54.4	52.4
	BERT-DRE	76.7	84.9	80.5	83.1	86.5	84.8	46.2	48.2	47.1
	DHG-BERT	80.8	85.5	**83.1**	83.5	88.0	**85.6**	57.3	60.2	**58.8**

Table 1: Results on the CDR and GDA datasets, bold marks highest number among all compared models, † refers the updated results on the official github repo.

($\mathrm{KBID} = -1$) are removed. Besides, we randomly initialize the type, coreference, and distance embeddings with 20-dimensions vectors. For CDR, the hidden dimension size of BiLSTM and the node vector dimension in DHG are both set to 256. The embedding size of BERT is 768, and a linear-transformation layer is utilized to project the BERT embedding into a low-dimensional space with the same size of node vectors. The model is trained using ADAM for 50 epochs with the initial learning rate of $3e^{-4}$. The layer number of SML and RRL are both set to 2, and all edges in DHG are undirected. For GDA, the only difference is that we set the initial learning rate to $1e^{-4}$. Beyond that, we run Stanford CoreNLP v3.9.2 (Manning et al., 2014) to generate dependency parse trees. All hyper-parameters are tuned according to the results on dev sets. We select the model with the median dev F1 from 5 independent runs and report its test F1, and F1 scores for intra- and inter- sentence entity pairs are also reported.

4.4 Main Results

Table 1 report the results of our proposed models against other baseline methods on two datasets. It can be observed that models with DHG significantly outperform all other approaches, and DHG-BERT achieves the state-of-the-art F1 score on all datasets. Compared with the latest graph-based methods EoG and GCNN on the CDR dataset, our DHG-LSTM achieves substantial improvements of 1.1% and 6.1% in F1 score, respectively. We attribute the performance gain to two design choices: (1) the decomposition of document modeling and multi-hop reasoning since it enables the reasoning process to benefit from the sequential and structural information; (2) the weighted mechanism in our message propagation strategy as it collects an adaptive amount of information from heterogeneous neighboring nodes.

Besides, DHG-LSTM improves by a relative margin of 6.5% against LSTM-DRE on the CDR dataset, and even though BERT already provides strong power of learning rich semantic features, DHG-BERT still achieves consistent improvement. It directly proves the necessity of incorporating structural information and reasoning mechanism in document-level RE, and many kinds of basic encoder could be well integrated into our model due to the loosely-coupled architecture. We consider the performance gain on BERT-DRE is mainly because that DHG makes full use of the semantic information of BERT, and the graph structure complements the weakness of BERT in capturing long-range syntactic structure, which is consistent with the conclusion of some recent studies (Clark et al., 2019; Zhang et al., 2020b).

Model	Dev F_1		
	Overall (%)	Intra (%)	Inter (%)
DHG-LSTM	64.9	69.3	54.5
– Context Encoding Layer (CEL)	59.7	64.0	49.4
– Structure Modeling Layer (SML)	60.2	64.5	50.3
– Relation Reasoning Layer (RRL)	62.6	68.0	48.8
– Dual-tier Structure	62.5	66.2	52.0
– Weighted Mechanism	64.2	68.7	53.2
– Shortcut Connection	63.3	67.5	52.8
– Input Augmentation	63.7	67.8	53.9
– Word-Word Adjacent (WWA)	63.1	67.4	52.2
– Word-Word Dependency (WWD)	64.0	68.3	53.5
– Sentence-Sentence Adjacent (SSA)	63.5	67.9	52.4
– Sentence-Sentence Complement (SSC)	62.6	66.7	51.1
– Sentence-Sentence (SSA & SSC)	59.3	65.3	32.7
– Mention-Mention Cooccurrence (MMO)	64.5	68.8	54.0
– Mention-Mention Coreference (MMR)	63.8	68.6	50.4
– Entity-Entity Complement (EEC)	62.9	67.5	50.2

Table 2: Results on CDR dataset with different model architectures (top) and edges (bottom). To prevent having disconnected graphs, we do not try to remove affiliation edges (i.e., WSF and MEF).

Meanwhile, another phenomenon is that the DHG-based models outperform all baseline models in both intra- and inter- sentence scenarios, especially the inter-sentence setting, which demonstrates that the majority of DHG mainly comes from inter-sentence relational facts. In particular, when comparing our DHG-LSTM with its baseline LSTM-DRE, one can find that there is a breathtaking improvement for the performance of inter-sentence pairs, and the intra-sentence pairs also substantially benefit from the well-organized document-level information.

4.5 Effect of Model Architectures

To study the contributions of different modules in our model, we run an ablation study on the CDR dataset (see the top part of Table 2). From these ablations, one can observe that: (1) CEL, SML, RRL are indispensable layers that bring 5.2%, 4.7%, and 2.3% improvements in F1 score to the ultimate performance, respectively, which suggests that they play different decisive roles in the entire system. (2) Without the dual-tier structure, the performance also suffers grievous damage by 2.4% F1. It is strong evidence that layering these heterogeneous nodes instead of desultorily mixing them up is quite pivotal. (3) The weighted mechanism contributes about 0.7% F1, indicating that it is necessary to let model aware of edge type such as complementary edges should be less weighted than others. (4) The operation of shortcut connection in SML is crucial since the F1 drops markedly by 1.6% if it is removed, which can be interpreted that the shortcut provides an effective way to combine sequential with structural information and tackles the vanishing gradient problem in deep neural networks. (5) Removing the input augmentation hurts the final result by 1.2% F1, which shows that the participation of multi-channel information can also help the document-level RE model improve performance.

4.6 Effect of Different Edges

In this experiment, we investigate the influence of different edges available in our DHG. For this purpose, we ablate each type of edges independently and report the results at the bottom of Table 2. The first thing to note is that the F1 score drops markedly when WWA or SSA is removed, which could be interpreted as they hold the sequential structure of a document, and one can restore the original document via these two edges. Secondly, WWD brings a remarkable improvement, which justifies the effectiveness of utilizing syntactic dependency in document-level RE. Thirdly, removing MMO slightly harms the performance as most of the intra-sentence relations could be identified using context encoding layer and a word-sentence-word chain in SML is capable of replacing part of its role. In contrast, the discard of MMR reduces the performance significantly since MMR enables the model to perceive long-term and

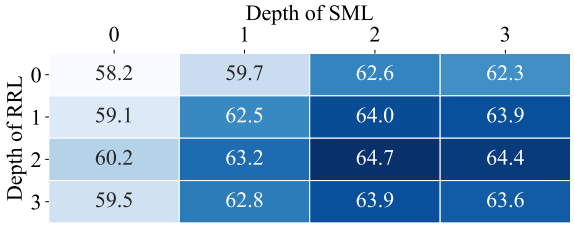

Figure 4: Results of DHG-LSTM on CDR dataset when using different number of layers, 0-layer indicates the module is not used.

inter-sentence mention coreference relations. Last but not least, SSC and EEC facilitate inter-sentence relation extraction and also play necessary roles to make information quickly propagate over the graph. Unsurprisingly, the removal of sentence-to-sentence connections (SSA & SSC) leads to a sharp deterioration of performance, especially the inter-sentence pairs. Overall, every edge performs its own duty. Various edges work in the mutual promotion way, which again confirms our motivation that explicitly capturing document inherent structure is essential for document-level RE.

4.7 Effect of Model Depth

We explore the impact of model depth (number of layers) in this section. For SML and RRL in the DHG-LSTM model, we vary their layer numbers from 0 to 3. As shown in Figure 4, the model reaches its optimal performance when the layers of SML and RRL are both 2. In such a circumstance, word nodes in SML perceive the information of all sentence nodes with the sentence-sentence-word chain, and RRL enables the 2-hop reasoning. Thus the key information for detecting relations could be fully aggregated. However, neither shallow model nor deep model works very well. One possible reason is that only collecting information from nearest neighbor nodes is not enough to identify the relation between two entities. In contrast, when the layer number equals 3, any two nodes in the same graph are accessible, which may introduce redundant information and hinder the inference.

5 Related Work

The study presented in this paper is directly related to existing researches on document-level relation extraction (Verga et al., 2018; Yao et al., 2019), which is recently introduced as a branch of relation extraction. The goal of this task is to identify all relations between each entity pair within a given document. Some early work first designs an extensive set of features and then trains a classifier based on these feature vectors (Xu et al., 2016; Zhou et al., 2016; Gu et al., 2017). Later, many neural network-based methods are introduced to solve the problem. Still, most of these approaches use sequential models, which can be regarded as simple extensions of sentence-level RE models (Nguyen and Verspoor, 2018; Li et al., 2018a). Beyond that, the approach presented in this paper is related to recent studies using graph neural networks for document-level relation extraction (Christopoulou et al., 2019; Sahu et al., 2019). Different from the previous work, our innovation lies in that we creatively propose to utilize a dual-tier heterogeneous graph to model the inherent structure of a document in the lower layer, and then enable multi-hop relational reasoning in the upper layer, instead of handling these complex requirements with one messy or imperfect graph. Meanwhile, the weighted mechanism in our message propagation strategy is inspired by the success of WGCN (Shang et al., 2019), which shows the effectiveness of using learnable weights to determine the amount of information from neighbors used in local aggregation. Besides these studies, our work is also relevant to the following research directions:

Sentence-level RE: Sentence-level is the most classical and simple setting of RE. Zhang et al. (2017) and Zhang et al. (2018) demonstrate that the positional and structural information is quite effective for sentence-level RE. There are also some works employ graph-based models for cross-sentence RE (Peng et al., 2017; Song et al., 2018; Gupta et al., 2019), but they ideally restrict all relation candidates in one continuous sentence-span with a fixed length, while the problem we want to solve in this paper requires extracting relations from the whole document, which is more difficult but practical.

GNNs for NLP: Recently, there is a considerable amount of interest in applying GNNs to NLP tasks. For example, in neural machine translation, GNNs have been employed to integrate syntactic and semantic information into encoders (Marcheggiani et al., 2018); De Cao et al. (2019), Cao et al. (2019) and Tu et al. (2019) employ GNNs over a heterogeneous graph to do multi-hop machine reading comprehension, which inspire our idea of the relation reasoning layer of DHG.

6 Conclusion

In this paper, we present a novel GNNs-based approach for document-level RE with DHG, a dual-tier heterogeneous graph, to achieve document modeling and multi-hop reasoning in proper order. Experimental results on two widely used document-level RE datasets suggest that the proposed model achieves state-of-the-art performance. We believe our approach is robust enough and can be readily adapted for other document-level NLP tasks without much manual efforts for domain adaptation. In the future, we would like to investigate explainable GNNs for document-level RE and integrate pre-training techniques with the proposed dual-tier heterogeneous structure in document-level NLP tasks.

Acknowledgements

This research is supported by the National Key Research and Development Program of China (grant No. 2016YFB 0801003) and the Strategic Priority Research Program of Chinese Academy of Sciences (grant No. XDC02040400).

References

Yu Cao, Meng Fang, and Dacheng Tao. 2019. Bag: Bi-directional attention entity graph convolutional network for multi-hop reasoning question answering. In *Proc. of NAACL*.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proc. of BioNLP@ACL*.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the dots: Document-level neural relation extraction with edge-oriented graphs. In *Proc. of EMNLP*.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What does bert look at? an analysis of bert's attention. In *Proc. of BlackboxNLP@ACL*.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question answering by reasoning across documents with graph convolutional networks. In *Proc. of NAACL*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL*.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *Proc. of ICML*.

Jinghang Gu, Fuqing Sun, Longhua Qian, and Guodong Zhou. 2017. Chemical-induced disease relation extraction via convolutional neural network. *Database*.

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, Bernt Andrassy, and Thomas A. Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *Proc. of AAAI*.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *Proc. of ICLR*.

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*.

Haodi Li, Ming Yang, Qingcai Chen, Buzhou Tang, Xiaolong Wang, and Jun Yan. 2018a. Chemical-induced disease extraction via recurrent piecewise convolutional neural networks. *BMC MED INFORM DECIS*.

Junjie Li, Haitong Yang, and Chengqing Zong. 2018b. Document-level multi-aspect sentiment classification by jointly modeling users, aspects, and overall ratings. In *Proc. of COLING*.

Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proc. of ACL*.

Diego Marcheggiani, Joost Bastings, and Ivan Titov. 2018. Exploiting semantics in neural machine translation with graph convolutional networks. In *Proc. of ACL*.

Lesly Miculicich, Dhananjay Ram, Nikolaos Pappas, and James Henderson. 2018. Document-level neural machine translation with hierarchical attention networks. In *Proc. of EMNLP*.

Dat Quoc Nguyen and Karin Verspoor. 2018. Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings. In *Proc. of BioNLP@ACL*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *TACL*.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Sunil Kumar Sahu, Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Inter-sentence relation extraction with document-level graph convolutional neural network. In *Proc. of ACL*.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne van den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *Proc. of ESWC*.

Chao Shang, Yun Tang, Jing Huang, Jinbo Bi, Xiaodong He, and Bowen Zhou. 2019. End-to-end structure-aware convolutional networks for knowledge base completion. In *Proc. of AAAI*.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proc. of ACL*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state lstm. In *Proc. of EMNLP*.

Ming Tu, Guangtao Wang, Jing Huang, Yun Tang, Xiaodong He, and Bowen Zhou. 2019. Multi-hop reading comprehension across multiple documents by reasoning over heterogeneous graphs. In *Proc. of ACL*.

Patrick Verga, Emma Strubell, and Andrew McCallum. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. In *Proc. of NAACL*.

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 *Proc. of RECOMB*.

Jun Xu, Yonghui Wu, Yaoyun Zhang, Jingqi Wang, Hee-Jin Lee, and Hua Xu. 2016. Cd-rest: a system for extracting chemical-induced disease relation in literature. *Database*.

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 *Proc. of ACL*.

Bowen Yu, Zhenyu Zhang, Tingwen Liu, Bin Wang, Sujian Li, and Quangang Li. 2019. Beyond word attention: using segment attention in neural relation extraction. In *Proc. of IJCAI*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proc. of EMNLP*.

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

Zhenyu Zhang, Xiaobo Shu, Bowen Yu, Tingwen Liu, Jiapeng Zhao, Quangang Li, and Li Guo. 2020a. Distilling knowledge from well-informed soft labels for neural relation extraction. In *Proc. of AAAI*.

Zhuosheng Zhang, Yuwei Wu, Junru Zhou, Sufeng Duan, Hai Zhao, and Rui Wang. 2020b. Sg-net: Syntax-guided machine reading comprehension. In *Proc. of AAAI*.

Huiwei Zhou, Huijie Deng, Long Chen, Yunlong Yang, Chen Jia, and Degen Huang. 2016. Exploiting syntactic and semantics information for chemical–disease relation extraction. *Database*.