Global Context-enhanced Graph Convolutional Networks for Document-level Relation Extraction

Huiwei Zhou, Yibin Xu, Zhe Liu, Weihong Yao*, Chengkun Lang, Haibin Jiang
School of Computer Science and Technology, Dalian University of Technology, China
{zhouhuiwei, weihongy}@dlut.edu.cn
{19xyb, njjnlz, kunkun, jianghaibin}@mail.dlut.edu.cn

Abstract

Document-level Relation Extraction (RE) is particularly challenging due to complex semantic interactions among multiple entities in a document. Among exiting approaches, Graph Convolutional Networks (GCN) is one of the most effective approaches for document-level RE. However, traditional GCN simply takes word nodes and adjacency matrix to represent graphs, which is difficult to establish direct connections between distant entity pairs. In this paper, we propose Global Context-enhanced Graph Convolutional Networks (GCGCN), a novel model which is composed of entities as nodes and context of entity pairs as edges between nodes to capture rich global context information of entities in a document. Two hierarchical blocks, Context-aware Attention Guided Graph Convolution (CAGGC) for partially connected graphs and Multi-head Attention Guided Graph Convolution (MAGGC) for fully connected graphs, could take progressively more global context into account. Meantime, we leverage a large-scale distantly supervised dataset to pre-train a GCGCN model with curriculum learning, which is then fine-tuned on the human-annotated dataset for further improving document-level RE performance. The experimental results on DocRED show that our model could effectively capture rich global context information in the document, leading to a state-of-the-art result. Our code is available at https://github.com/Huiweizhou/GCGCN.

1 Introduction

The task of Relation Extraction (RE) aims to detect semantic relations among entities in text, which plays an important role in many natural language processing applications such as knowledge discovery (Quirk and Poon, 2017), and question answering (Yih et al., 2015; Yu et al., 2017).

Previous research on relation extraction mainly focuses on sentence-level, i.e., predicting relations between entity pairs in a given sentence. However, in real-world scenarios, many relations are expressed across sentences. The task of identifying these relations is named inter-sentence RE. Typically, inter-sentence relations occur in textual snippets with several sentences, such as documents. In a document, multiple mentions of the target entities in different sentences should be used for inter-sentence relation extraction, since their relations are expressed through the interactions of these mentions in the whole document.

Yao et al. (2019) introduce a dataset called DocRED to accelerate the research on document-level RE. Take a document from DocRED as an example in Figure 1.There are many simple intra-sentence relations, such as ("*Ikwilalles met je delen*", entry song, *Eurovision Song Contest 1990*) in sentence 1, ("*Ikwilalles met je delen*", performed after, "*Quand je terêve"*) and (*Céline Carzo*, performer, "*Quand je terêve"*) in sentence 5. Only one evidence sentence is needed to predict these relation facts. However, the inter-sentence relation (*Céline Carzo,* participant of*, Eurovision Song Contest 1990*) is supported by two evidence sentences (sentence 1 and sentence 5), and should be inferred based on the above three

* Corresponding author

Proceedings of the 28th International Conference on Computational Linguistics, pages 5259–5270
Barcelona, Spain (Online), December 8-13, 2020

Input:

[1] "Ikwilalles met je delen" ("I want to share everything with you") was the Dutch entry in the ***Eurovision Song Contest 1990***, performed in Dutch by Maywood. [2] The English language version was entitled "No more winds to guide me". [3] The song is a ballad, with the singer telling her lover that she wants to share everything with him-including the hard times in life. [4] She sings that... [5] The song was performed fifth on the night, following Turkey's Kayahan with Gözlerinin Hapsindeyim and preceding Luxembourg's ***Céline Carzo*** with "Quand je terêve". [6] At the close of voting, it had received 25 points, placing 15th in a field of 22...

Subject: ***Céline Carzo***

Object: ***Eurovision Song Contest 1990***

Relation: ***participant of*** **Supporting Sentence:** 1, 5

Figure 1: An example from DocRED. Each document in DocRED is annotated with named entity mentions, coreference information, and supporting sentences.

intra-sentence relations. Since inter-sentence relations are inferred based on multiple relations, they are also called multi-hop relations. The prediction of inter-sentence relations is much more difficult than that of intra-sentence relations.

Previous work in document-level RE employ hierarchical inference networks or Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) to extract features from local level to global level for multi-hop relational reasoning (Wang et al., 2019; Kim et al., 2020; Guo et al., 2019; Sahu et al., 2019). How to construct a hierarchical inference network with GCN to encode rich global context information is crucial for document-level RE.

In this paper, we propose novel Global Context-enhanced Graph Convolutional Networks (GCGCN) with entities as nodes and context of entity pairs as edges between nodes for document-level RE. GCGCN is composed of two hierarchical inference blocks. The first block Context-aware Attention Guided Graph Convolution (CAGGC) connects two entities if they co-occur in at least one sentence. All sentences where a pair of entities co-occur are represented as an edge between the two entity nodes. Thus, CAGGC can learn an entity node representation based on context representations of all its mentions in the document and its neighbour nodes to encode local and global information. The second block Multi-head Attention Guided Graph Convolution (MAGGC) applies multi-head attention (Vaswani et al., 2017) to generate multiple fully connected edge-weighted graphs. MAGGC aims to enhance global context representations by connecting entity pairs in different sentences for multi-hop relational reasoning.

Furthermore, to reliably estimate the parameters of GCGCN model, we introduce a large-scale distantly supervised dataset in DocRED with curriculum learning to pre-train our model, and then fine-tune it on the human annotated dataset for improving the performance.

In summary, we mainly make the following contributions:

- We propose novel Global Context-enhanced Graph Convolutional Networks (GCGCN) with entities as nodes and context of entity pairs as edges between nodes to capture rich global context information.
- Two hierarchical inference blocks, CAGGC for partially connected graphs and MAGGC for fully connected graphs, could take progressively more global context into account.
- We further adopt curriculum learning to pre-train our model on a large-scale distantly supervised dataset to achieve better performance. Experiments on DocRED show that our model could capture complex semantic interactions across all entities in the document for multi-hop relational reasoning.

2 Related work

There is considerable research effort in the document-level RE task. Wang et al. (2019) apply BERT to encode the document for better capturing context information. Tang et al. (2020) propose a Hierarchical Inference Network (HIN), which can aggregate inference information from entity-level to sentence-level and then to document-level. Kim et al. (2020) extract global level relations from a document by utilizing the knowledge graph constructed from local relations. It is important to note that the hierarchical inference mechanism from local level to global level is necessary for document-level RE.

In recent years, Graph Convolutional Networks (GCN) (Kipf and Welling, 2017) have attracted much attention in natural language processing (Marcheggiani and Titov, 2017; Schlichtkrull et al., 2018; Cao et al., 2019), and have been approved effective for modelling sentence-level and document-level RE.

The most existing sentence-level GCN for relation extraction are built on dependency structures over input sentences. These methods construct graphs with words as nodes and dependency relations as edges between nodes (Zhang et al., 2018; Mandya et al., 2020). Instead of dependency structures, Zhu et al. (2019) construct a fully connected graph on unstructured texts with entities as nodes, and propagate relational information among nodes for multi-hop relational reasoning.

As for document-level RE, Guo et al. (2019) use Attention Guided GCN to transform the original dependency tree into a fully connected edge-weighted graph for encoding relations across sentences. Besides syntactic dependency edges, Sahu et al. (2019) introduce inter-sentence dependencies, such as coreference edges, adjacent sentence edges etc., into a document-level graph for inter-sentence relation extraction. Christopoulou et al. (2019) construct a document-level graph with mention, entity and sentence as nodes, and dependencies between these nodes as edges to infer entity-to-entity relations.

3 Graph Construction

In this section, we will introduce how to construct a graph for each document as an input of graph convolutional networks.

For every document, there is a set of entities denoted as $\mathbf{E} = \{e_v\}_{v=1}^{N}$, where N is the total number of entities, each entity e_v may contain multiple mentions $\{\mathrm{m}_i\}_{i=1}^{M}$. We construct the graph $G(\mathbf{A}, \mathbf{E})$ according to the following rules, where $\mathbf{A}$ is the adjacency matrix.

(1) We treat each entity in $\mathbf{E}$ as a node in the graph, that is to say $\mathbf{E}$ is the node set of the graph.

(2) If two entities (nodes) co-occur in the same sentence, we will build an edge between them. Since an entity pair can appear in different sentences, an edge may correspond to multiple sentences.

(3) By analyzing the training data, we find that in two adjacent sentences, the pronouns in the latter sentence often refer to the entities in the former one. To include pronouns and their referring entities in one sentence, we simply concatenate the two adjacent sentences if the second sentence contains pronouns such as "it". By this way, we can directly connect interacting entities in the two sentences by an edge. We call this strategy *Extend graph*. The pronoun list is set by manual statistics in advance. Specially, we tag POS of each word in the training set and select the most frequent pronouns, which are also provided at https://github.com/Huiweizhou/GCGCN with the code.

4 Global Context-enhanced Graph Convolutional Networks

Global Context-enhanced Graph Convolutional Networks (GCGCN) has four main components: an encoder layer, a Context-aware Attention Guided Graph Convolution (CAGGC) block, a Multi-head Attention Guided Graph Convolution (MAGGC) block and a classification layer, as shown in Figure 2.

4.1 Encoder layer

The encoder layer first encodes a given document matrix $\mathbf{D} = (w_{1,1}, w_{1,2}, ..., w_{i,j}, ..., w_{m,N_m})$ into a hidden vector sequence:

$$\mathbf{D}' = \mathrm{Encoder}(w_{1,1}, w_{1,2}, ..., w_{i,j}, ..., w_{m,N_m}) = (h_{1,1}, h_{1,2}, ..., h_{i,j}, ..., h_{m,N_m}) \quad (1)$$

where Encoder is BiLSTM or BERT, $h_{i,j} \in \mathbb{R}^d$ is the representation of *j*-th word in *i*-th sentence, *d* is the dimension of hidden representations.

We then compute each node representation in the graph. Since there may be multiple mentions of the same entity, we perform average operations on them to obtain the entity representation. Specifically, for each mention m_k ranging from *s*-th word to *t*-th word in *i*-th sentence corresponding to entity e_v, we compute the mention representation as $R_k = \frac{1}{t-s+1}\sum_{j=s}^{t} h_{i,j}$, and then the representation of entity e_v is calculated as $\mathrm{P}_v = \frac{1}{J}\sum_{k=1}^{J} R_k$, where J is the number of mentions for e_v. We denote the initial node representations in the encoder layer as $\mathbf{P}^{(0)}$.

4.2 Context-aware Attention Guided Graph Convolution (CAGGC)

CAGGC is the first block in our model that used to create a partially connected graph. Different from traditional GCN, our model considers not only node representations, but also edge representations in graph construction.

Entity-aware edge representations: In order to obtain the edge representations between the nodes *u* and *v*, which may correspond to more than one sentence, we perform a word-level attention mechanism to get sentence representations and a gate mechanism to get entity-aware edge representations. Specifically, we take advantage of every word embedding and relative distance to the given entity to calculate the representation h_i for *i*-th sentence on edge *uv*, the process is as follows:

$$\alpha_{i,j}^{c} = \mathrm{softmax}\left(z^{\mathrm{T}} \tanh\left(\mathbf{W}_1 h_{i,j} + \mathbf{W}_2 \mathrm{x}_{pos(i,j)}^{c} + b_1\right)\right) \tag{2}$$

$$h_i^c = \sum_{j=1}^{m} \alpha_{i,j}^{c} h_{i,j} \tag{3}$$

where $c \in \{u, v\}$ represents any one of the two entities, $\mathrm{x}_{pos(i,j)}^{c}$ is the relative distance of the current word to entity *c*, $\alpha_{i,j}^{c}$ is the attention weight of every word *j* in sentence *i* with respect to entity *c*, *m* is the number of words in *i*-th sentence, $\mathbf{W}_1$, $\mathbf{W}_2$, z and b_1 are all trainable parameters. For simplicity, hereafter we will not explain trainable parameters $\mathbf{W}$ and b in equations.

For entity *u* and *v*, we perform the attention operation respectively, and get two representations h_i^u and h_i^v for *i*-th sentence. Then they are concatenated and fed to a full connection layer to get the representation of *i*-th sentence $h_i = \mathbf{W}_{wa}[h_i^u; h_i^v] + b_{wa}$.

Next, inspired by Li et al. (2020), a gate mechanism is applied to obtain edge representations, which allows the model jointly attends to information from all sentences in edge *uv*. For each entity node $c \in \{u, v\}$, its representation $\mathrm{P}_c^{(0)}$ is used to calculate a weighted sum of all the sentence representations on edge *uv* as follows:

$$\beta_i^c = \sigma\left(\mathbf{W}_3^{\mathrm{T}} \tanh\left(\mathbf{W}_4 h_i + \mathbf{W}_5 \mathrm{P}_c^{(0)} + b_2\right)\right) \tag{4}$$

$$\hat{h}_{u,v}^{c\ (1)} = \frac{1}{S}\sum_{i=1}^{S} \beta_i^c h_i \tag{5}$$

where $\sigma(\cdot)$ is sigmoid or ReLU activation function.

Two representations $\hat{h}_{u,v}^{u\ (1)}$ and $\hat{h}_{u,v}^{v\ (1)}$ are concatenated and fed to a full connection layer to produce entity-aware edge representations as $\hat{h}_{u,v}^{(1)} = \mathbf{W}_{sg}[\hat{h}_{u,v}^{u\ (1)}; \hat{h}_{u,v}^{v\ (1)}] + b_{sg}$. Thus, we get edge representation matrix $\hat{\mathbf{H}}^{(1)}$ for CAGGC.

Note that our gate mechanism has two characteristics. First, it introduces the representations of two entities to calculate the gate value, which gives larger weight to sentences related to the two entities.

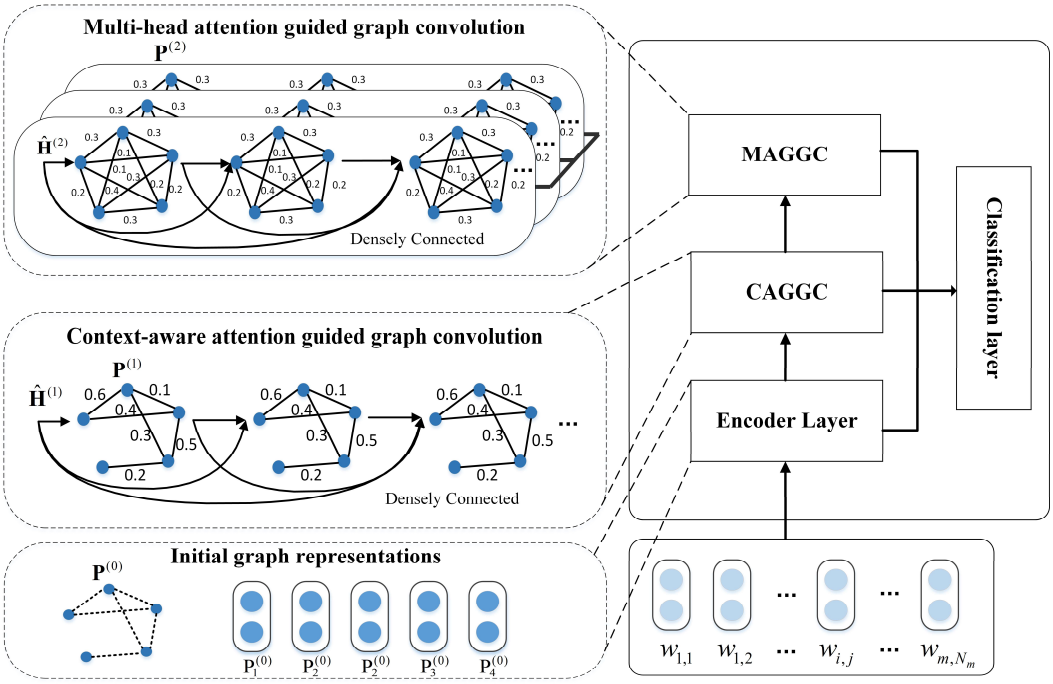

Figure 2: The overall architecture of GCGCN. The GCGCN model is shown with an example document which has 5 entities. Firstly, an encoder layer is used to generate initial entity representations. Then, with the help of two hierarchical inference blocks, our model can learn rich global information in the graph. Finally, a classification layer is used to concatenate the representations obtained in the encoder layer and two blocks, and predicts the relations between entities.

Second, the activation function is used to calculate the weight of the sentence, so that our model can effectively control the information flows even there is only one sentence on edge.

Weighted adjacency matrix: The adjacency matrix used in traditional GCN is composed of 0 and 1 to indicate whether there are edge connections between nodes, which cannot effectively control the information propagation between entities. We propose a novel method for calculating the weighted adjacency matrix, which comprehensively considers the information of nodes and edges. The weight between nodes u and v is denoted as $\mathrm{A}_{u,v}^{(1)}$, which can be computed as follows:

$$\mathrm{A}_{u,v}^{(1)} = \frac{\exp\left(\tanh\left(\mathbf{W}^{\mathrm{T}}\left(\mathbf{W}_u \mathrm{P}_u^{(0)} + \mathbf{W}_v \mathrm{P}_v^{(0)} + \mathbf{W}_e \hat{h}_{u,v}^{(1)}\right)\right)\right)}{\sum_{u \in neighbour(v)} \exp\left(\tanh\left(\mathbf{W}^{\mathrm{T}}\left(\mathbf{W}_u \mathrm{P}_u^{(0)} + \mathbf{W}_v \mathrm{P}_v^{(0)} + \mathbf{W}_e \hat{h}_{u,v}^{(1)}\right)\right)\right)} \tag{6}$$

Graph convolution operation: The edge information is also introduced to the graph convolution operation to take advantage of rich context information to update node representations. Every block in our model contains K densely connected sublayers. For node v, its representation at the k-th sublayer is calculated as:

$$\mathrm{P}_v^k = \mathrm{ReLU}\left(\sum_{u \in neighbour(v)} \mathrm{A}_{u,v}^{(1)}\left(\mathbf{W}_{node}^k \tilde{\mathrm{P}}_u^{k-1} + \mathbf{W}_{edge}^k \hat{h}_{u,v}^{(1)} + b^k\right)\right) \tag{7}$$

where $\mathbf{W}_{node}^k$, $\mathbf{W}_{edge}^k$ and b^k is the trainable parameters of k-th sublayer.

In order to combine the output representations of all proceeding k-1 sublayers which is denoted as $\tilde{\mathrm{P}}_u^{k-1}$, dense connections (Huang et al., 2017) are used in our model:

$$\tilde{\mathrm{P}}_u^{k-1} = [\mathrm{P}_u^{(0)}; \mathrm{P}_u^1; \ldots; \mathrm{P}_u^{k-1}] \tag{8}$$

To combine these representations without changing the output node dimensions, a linear layer is used to reduce dimensions of them, the dimensions of these sublayers d_{hidden} are computed as $d_{hidden}=d/k$, where d is the input feature dimension.

We apply dense connections operation on initial node representations $\mathbf{P}^{(0)}$ and then concatenate the output representations of all K sublayers to form the new node representations $\mathbf{P}^{(1)} = [\mathbf{P}^{(0)}; \mathbf{P}^{1}; ...; \mathbf{P}^{K}]$, which are fed to the next block.

With the help of dense connections, our model is going deeper, capturing rich local and global context information for a better graph representation.

4.3 Multi-head Attention Guided Graph Convolution (MAGGC)

Traditional GCN based RE models can only establish connections between directly connected or close entities. To solve this problem, we introduce Attention Guided GCNs (Guo et al., 2019) to our model at MAGGC block. It can collect interactions among all nodes by using multi-head attention, especially for those connected by multi-hop paths.

Attention guided layer: In attention guided layer, the partially connected graph constructed in the first block is transformed into a fully connected edge-weighted graph. The same method (Equation (4-5)) as the first block is used to calculate entity-aware edge representations $\hat{\mathbf{H}}^{(2)}$ with node representations $\mathbf{P}^{(1)}$. If two entities u and v do not appear in the same sentence, we represent the edge $\hat{h}_{u,v}^{(2)}$ as a zero vector.

Instead of considering the impact of contextual information as that in CAGGC, we compute adjacency matrix $\mathbf{A}^{(2)}$ for MAGGC by using self-attention mechanism (Vaswani et al., 2017), which is formally denoted as:

$$\mathbf{A}^{(2)} = \text{softmax}\left(\frac{(\mathbf{W}_Q\mathbf{P}^{(1)})^T(\mathbf{W}_K\mathbf{P}^{(1)})}{\sqrt{d}}\right) \tag{9}$$

Multi-head attention: Inspired by the multi-head attention (Vaswani et al., 2017), we use the above formula to calculate t different adjacency matrices $\{\mathbf{A}_1^{(2)}, \mathbf{A}_2^{(2)}, ..., \mathbf{A}_t^{(2)}\}$. Then, we take advantage of each calculated adjacency matrix, edge representations $\mathbf{H}^{(2)}$ and the node representations $\mathbf{P}^{(1)}$ to perform the graph convolution operation as Equation (7-8), respectively. Next, we concatenate all t output representation $\{\mathbf{P}_1^{(2)}; \mathbf{P}_2^{(2)}; ...; \mathbf{P}_t^{(2)}\}$ together and apply a transform operation to reduce the dimension. Finally, we get the node representations $\mathbf{P}^{(2)}$, which is the same size as the initial node representations.

4.4 Classification layer

We concatenate the node representations of two blocks with the initial node representations of the encoder layer, and get the final node representations $\mathbf{P}$ through the processing of a full connection layer, which can be formulized as:

$$\mathbf{P} = \tanh(\mathbf{W}_P[\mathbf{P}^{(0)}; \mathbf{P}^{(1)}; \mathbf{P}^{(2)}] + b_P) \tag{10}$$

where $\mathbf{P}^{(0)}$ is the initial representation, $\mathbf{P}^{(1)}$ and $\mathbf{P}^{(2)}$ are the output representations of CAGGC and MAGGC. $\mathbf{W}_P$ and b_P are the trainable parameters.

Entity types (e.g., PRE, LOC, ORG) and relative distances are also used to enrich entity representations. Entity types and relative distances are mapped to entity type embeddings and relative distance embeddings by the entity type embedding matrix and the relative distance embedding matrix, respectively. In practice, for an entity pair (e_u, e_v), we concatenate each node representation obtained by graph convolution process with its entity type embedding and relative distance embedding, which are then fed to a bilinear function and a fully connected layer to obtain the relation feature for relation prediction. The procedure is formalized as:

$$\mathrm{P}'_u = [\mathrm{P}_u; t_u; d_{u,v}] \tag{11}$$

$$\mathrm{P}'_v = [\mathrm{P}_v; t_v; d_{v,u}] \tag{12}$$

$$P(r|u,v) = \text{sigmoid}(\mathrm{P}'^{\mathrm{T}}_u\mathbf{W}_r\mathrm{P}'_v + \mathbf{W}_l[\mathrm{P}'_u; \mathrm{P}'_v] + b_r) \tag{13}$$

where $[\cdot;\cdot]$ denotes concatenation operation, t_u and t_v are type embeddings for entity u and v, $d_{u,v}$ and $d_{v,u}$ are relative distance embeddings of the first mentions of the two entities in the document.

The relation prediction in our task is a multi-label classification problem. During training, we take the binary cross entropy as loss function:

$$Loss = -\sum_{D \in S} \sum_{u \neq v} \sum_{r_i \in R} \mathrm{II}(r_i = 1) \log P(r_i|u,v) + \mathrm{II}(r_i = 0) \log(1 - P(r_i|u,v)) \tag{14}$$

where S denotes the whole corpus, $\mathrm{II}(\bullet)$ refers to indication function, and R is a pre-defined relation type set.

5 Experiments

5.1 Datasets and Evaluation Metrics

We evaluate our model on DocRED (Yao et al., 2019), which contains 3,053 documents for training, 1,000 for development and 1,000 for test, totally with 132,375 entities, 56,354 relational facts and 96 frequent relation types. It is also introduced by the author of DocRED that about 40.7% of relational facts can only be extracted from multiple sentences and 61.1% relational instances require a variety of reasoning. Along with the human-annotated dataset, a large-scale distant supervised dataset which contains 101,873 documents is also been provided.

The evaluation on test set is done through CondaLab[1]. The widely used metric F_1 is used in our experiments. Considering such a situation that some relational facts present in both the training and dev/test sets, a model may memorize their relations during training and achieves a better performance on dev or test set in an undesirable way. We also report the F_1 excluding those relational facts and denote it as Ign F_1.

5.2 Implementation Details

We set the number of densely connected sublayers K in CAGGC and MAGGC to 4, the number of heads t in MAGGC to 4. We use Adam with weight decay 0.0001 for optimization, and set the dropout rate to 0.2, the learning rate to 5e-6. The batch size is set to 1 because the graph convolution operation containing edge representations consumes a lot of memory. Our model is developed by Pytorch.

Two settings, **GCGCN-GloVe** and **GCGCN-BERT**, are implemented for our GCGCN. **GCGCN-GloVe** uses GloVe (100d) and BiLSTM (128d) as word embedding and encoder. **GCGCN-BERT** uses BERT-Base as encoder. The word representations of BERT-Base are mapped to 128d by a linear projection layer. The embedding dimensions of distance and entity type are all set to 20.

5.3 Main Results

We compare our proposed GCGCN model against some state-of-the-art document-level RE models on DocRED dataset, and show the main results in Table 1. We divide these models into four groups.

From the results. we can see that: (1) Among the GloVe-based and BERT-based models, the hierarchical inference models are generally better than the other models, which verifies that hierarchical inference methods could distinguish crucial entity-level, sentence-level and document-level inference information for overall document-level relational reasoning. Simply encoding the document cannot effectively model complex relationships between entities. (2) BERT-based models show a great improvement over GloVe-based models, which indicates that BERT is a powerful context encoder to model entity relations. (3) Both **GCGCN-GloVe** and **GCGCN-BERT** consistently achieves the best performance on the two groups. It proves that our GCGCN model could enhance global context representations with two blocks for document-level relational reasoning.

[1] https://competitions.codalab.org/competitions/20717

	Model	Dev		Test	
		Ign F_1(%)	F_1(%)	Ign F_1(%)	F_1(%)
1	CNN (Yao et al., 2019)	41.58	43.45	40.33	42.26
	LSTM (Yao et al., 2019)	48.44	50.68	47.71	50.07
	BiLSTM (Yao et al., 2019)	48.87	50.94	48.78	51.06
	ContextAware (Yao et al., 2019)	48.94	51.09	48.40	50.70
2	GREG-Context (Kim et al., 2020)	-	-	-	52.88
	HIN-GloVe (Tang et al., 2020)	51.06	52.95	51.15	53.30
	GCGCN-GloVe	51.14	53.05	50.87	53.13
3	BERT-RE (Wang et al., 2019)	-	54.16	-	53.20
4	BERT-Two-Step (Wang et al., 2019)	-	54.42	-	53.92
	HIN-BERT (Tang et al., 2020)	54.29	56.31	53.70	55.60
	GCGCN-BERT	**55.43**	**57.35**	**54.53**	**56.67**

Table 1: Performance of different models on DocRED. 1: GloVe-based models without hierarchical inference; 2: GloVe-based models with hierarchical inference; 3: BERT-based models without hierarchical inference; 4: BERT-based models with hierarchical inference.

5.4 Ablation Study

To understand the effect of different components, we conduct an ablation study on the dev set as illustrated in Table 2. Form the results, we can see that all components are effective in improving the performance.

Without introducing edge representations to calculate adjacency matrices in CAGGC and update node representations in CAGGC and MAGGC, hurts the result by 1.5%. This shows that edge representations can provide rich global context information for entity nodes.

If we replace the gate mechanism with the attention mechanism in edge representation computing, F_1 drops by 1.33%. The gate mechanism can selectively extract entity-related sentence representations with the given entities.

When we use traditional GCN to replace CAGGC or MAGGC, F_1 score drops by 1.14% and 1.22% respectively. Compared with traditional GCN, our two blocks have stronger ability to capture complex semantic interactions among multiple entities for document-level RE.

Extend graph contributes 1.09% F_1 score. Although simple, our strategy is effective for inter-sentence relational inference.

Setting	Ign F_1(%)	F_1(%)
GCGCN-BERT	**55.43**	**57.35**
w/o Edge representations	53.99 (↓1.44)	55.85 (↓1.50)
w/o gate w/ attention	54.09 (↓1.34)	56.02 (↓1.33)
w/o CAGGC w/ GCN	54.28 (↓1.15)	56.21 (↓1.14)
w/o MAGGC w/ GCN	54.20 (↓1.23)	56.13 (↓1.22)
w/o *Extend graph*	54.16 (↓1.27)	56.26 (↓1.09)
w/o Dense connections	53.28 (↓2.15)	55.27 (↓2.08)

Table 2: Ablation Study of GCGCN on DocRED dev set.

Removing dense connections between each sublayer, F_1 drops by 2.08%. Deeper GCNs can capture richer neighborhood information of a graph. However, as the number of sublayers increases, the over-smooth problem will occur. That is, introducing too much information from the whole graph makes node representations become similar and indistinguishable, which seriously affects GCN performance. Different from feed-forward connections, dense connections concatenate the representations of each sublayer together, which could extract multi-level features simply and efficiently. With the help of dense

connections, our GCGCN can learn better graph representations with rich local and global context information.

5.5 Analysis by the number of evidence sentences

To verify the effectiveness of our GCGCN model, especially the prediction ability of inter-sentence relations, we analyse the recall on relational facts with different number of evidence sentences and show results in Figure 3.

It can be seen that our GCGCN always performs best on both simple intra-sentence relation prediction and complex inter-sentence relation prediction compared to other baselines. To further show the functions of the two blocks clearly, we removed CAGGC and MAGGC from GCGCN separately to learn two corresponding single block models GCGCN-CAGGC and GCGCN-MAGGC. It can be observed that for simple intra-sentence relation prediction (0-1 evidence sentence), GCGCN-MAGGC perform better than BERT and GCGCN-CAGGC, which indicates that the block CAGGC could leverage local information to efficiently predict simple intra-sentence relations. While for complex inter-sentence relation prediction, GCGCN-CAGGC generally performs better than BERT and GCGCN-MAGGC, which demonstrates that MAGGC could obtain rich global information to help predict complex inter-sentence relations.

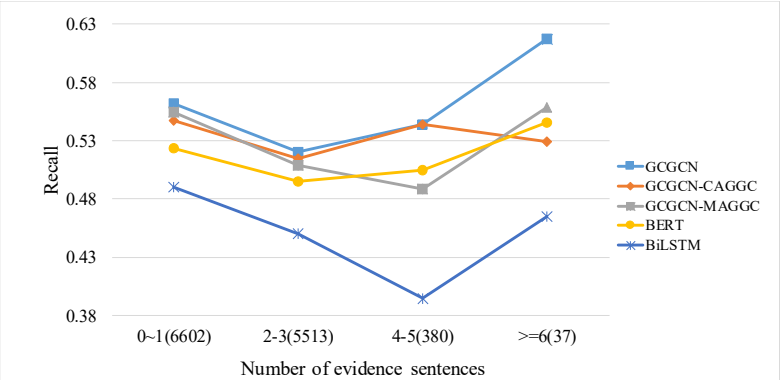

Figure 3: Recall of relation facts with different number of evidence sentences in dev set. The number in brackets is the number of relational facts with this amount of evidence sentences.

5.6 Effects of leveraging distantly supervised data

DocRED also offers a large-scale distantly supervised dataset. We introduce it to further improve the performance of our model. Due to the large amount of noise in the distantly supervised data, we do not directly add it to the human annotated dataset. It is used to pre-train our GCGCN model, which are then fine-tuned on the human-annotated dataset. The results are shown in Table 3. With the large-scale distantly supervised data, our model achieves a significant improvement in F_1.

To reduce the influence of the noisy data in the distantly supervised dataset, the curriculum learning strategy (Bengio et al., 2009) is applied to get a better pre-trained model. Different from the conventional training strategy from simple data to complex data, we believe that first training on low-quality data with more noise and then training on high-quality data with less noise will help improve the performance of the model.

We rank all documents in the distantly supervised dataset in order of high noise to low noise. Specifically, a GCGCN model is trained on the high quality human-annotated dataset, which is then used to predict entity labels L'_a of each document in the distantly supervised dataset. Next, we calculate an F_1 score for each document with the distantly supervised labels L_a and the corresponding L'_a. We consider that the higher the document F_1 score is, the more correct its labels are, and the less noisy labels it contains. Finally, all documents are ranked according to their F_1 scores from low to high for pre-training our model. With the help of curriculum learning, we finally get 62.39% F_1 score on test set.

Besides, we can also see that compared with F_1 score, Ign F_1 on dev/test sets of pre-train model improves very little. This shows that the improvement mainly comes from the overlap entity pairs between the large-scale distantly supervised dataset and dev/test sets.

Model	Dev		Test	
	Ign F_1(%)	F_1(%)	Ign F_1(%)	F_1(%)
GCGCN	55.43	57.35	54.53	56.67
GCGCN with pre-training	55.04	62.07	54.81	62.31
GCGCN with CL and pre-training	**56.07**	**62.49**	**54.89**	**62.39**

Table 3: Results of leveraging distantly supervised data and curriculum learning.

5.7 Case study

We compare our model with BiLSTM model on a sample from dev set in Figure 4. Our model correctly predicts multi-hop relation fact (*John Caselberg*, date of death, 2004), while BiLSTM cannot. This inter-sentence relation is inferred by the facts that *John Caselberg* is *Caselberg*'s husband from sentence 1 and 4, *Caselberg* died in later *2004*, and her husband died six months before her from sentence 6. The results demonstrate the ability of our model on multi-hop relational reasoning. With the help of hierarchical neural networks, GCGCN can collect inference information from the whole document and well predict inter-sentence relations.

Contrary to our expectation, GCGCN fails to predict the relation (*Edith Winifred Woollaston*, child, *Caselberg*) while BiLSTM can. We argue that *Caselberg* has complex semantic interaction with other entities, global information from the whole document brings an undesirable impact on local information.

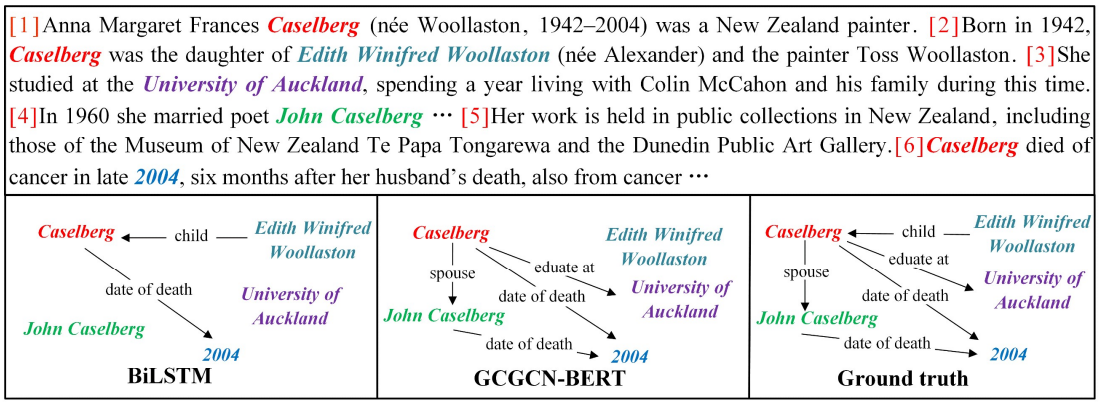

Figure 4: Sample predictions from BiLSTM model and our GCGCN-BERT model.

6 Conclusion

In this paper, we propose Global Context-enhanced Graph Convolutional Networks (GCGCN) to address the problem of document-level relation inference. Our model introduces context of entity pairs as edges between entity nodes to model complex semantic interactions among multiple entities in the document. The experiments on DocRED demonstrate that our model outperforms most existing models with an F_1 score of 62.39%. As further work, we would like to improve the inference mechanism and focus on the weakly supervised document-level RE.

Reference

Angrosh Mandya, Danushka Bollega, and Frans Coenen. 2020. Contextualised Graph Attention for Improved Relation Extraction. arXiv:2004.10624.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and IlliaPolosukhin. 2017. Attention Is All You Need. In *Conference and Workshop on Neural Information Processing Systems (NeurIPS)*.

Chris Quirk and Hoifung Poon. 2017. Distant Supervision for Relation Extraction beyond the Sentence Boundary. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*: Volume 1, Long Papers, pages 1171–1182, Valencia, Spain. Association for Computational Linguistics.

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.

Daniil Sorokin and Iryna Gurevych. 2017. Context-Aware Representations for Knowledge Base Relation Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

Diego Marcheggiani and Ivan Titov. 2017. Encoding Sentences with Graph Convolutional Networks for Semantic Role Labeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1506–1515, Copenhagen, Denmark. Association for Computational Linguistics.

Fenia Christopoulou, Makoto Miwa, and Sophia Ananiadou. 2019. Connecting the Dots: Document-level Neural Relation Extraction with Edge-oriented Graphs. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 4925–4936, Hong Kong, China.Association for Computational Linguistics.

Gao huang, Zhuang Liu, Laurens van der Maaten, Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261-2269.

Hao Zhu, Yankai Lin, Zhiyuan Liu, Jie Fu, Tat-seng Chua, and Maosong Sun. 2019. Graph Neural Networks with Generated Parameters for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1331-1339, Florence, Italy. 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. arXiv:2003.12754.

Hong Wang, ChristfriedFocke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune Bert for DocRED with Two-step Process. arXiv:1909.11898.

Kuekyeng Kim, YunaHur, Gyeongmin Kim, and Heuiseok Lim. 2020. GREG: A Global Level Relation Extraction with Knowledge Graph Embedding. *Applied Sciences*, 10(3):1181.

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 *European Semantic Web Conference 2018*, pages 593-607.

Mo Yu, Wenpeng Yin, KaziSaidul Hasan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2017. Improved Neural Relation Detection for Knowledge Base Question Answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics* (Volume 1: Long Papers), pages 571-581, Vancouver, Canada. Association for Computational Linguistics.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2019. Question Answering by Reasoning Across Documents with Graph Convolutional Networks. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*: Human Language Technologies, Minneapolis, USA, pages 2306–2317. Association for Computational Linguistics.

Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 756–765, Berlin, Germany, August 7-12, 2016. The Association for Computer Linguistics.

Sepp Hochreiter and Jurgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735– 1780.

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.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification With Graph Convolutional Networks. In *Proceedings of ICLR 2017*.

Wen-tau Yih, Ming-Wei Chang, Xiaodong He and Jianfeng Gao. 2015. Semantic Parsing via Staged Query Graph Generation: Question Answering with Knowledge Base. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1321–1331, Beijing, China. Association for Computational Linguistics.

Yang Li, Guodong Long, Tao Shen, Tianyi Zhou, Lina Yao, Huan Huo, and Jing Jiang. 2020. Self-Attention Enhanced Selective Gate with Entity-Aware Embedding for Distantly Supervised Relation Extraction. arXiv:1911.11899.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of 2009 International Conference on Machine Learning*, ACM.

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.

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.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention Guided Graph Convolutional Networks for Relation Extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, Florence, Italy. Association for Computational Linguistics.