frontiers in Genetics

ORIGINAL RESEARCH
published: 10 February 2021
doi: 10.3389/fgene.2021.624307

CID-GCN: An Effective Graph Convolutional Networks for Chemical-Induced Disease Relation Extraction

Daojian Zeng[1]*, *Chao Zhao*[2] and *Zhe Quan*[3]

[1] Hunan Provincial Key Laboratory of Intelligent Computing and Language Information Processing, Hunan Normal University, Changsha, China, [2] School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, China, [3] College of Information Science and Engineering, Hunan University, Changsha, China

Automatic extraction of chemical-induced disease (CID) relation from unstructured text is of essential importance for disease treatment and drug development. In this task, some relational facts can only be inferred from the document rather than single sentence. Recently, researchers investigate graph-based approaches to extract relations across sentences. It iteratively combines the information from neighbor nodes to model the interactions in entity mentions that exist in different sentences. Despite their success, one severe limitation of the graph-based approaches is the over-smoothing problem, which decreases the model distinguishing ability. In this paper, we propose CID-GCN, an effective Graph Convolutional Networks (GCNs) with gating mechanism, for CID relation extraction. Specifically, we construct a heterogeneous graph which contains mention, sentence and entity nodes. Then, the graph convolution operation is employed to aggregate interactive information on the constructed graph. Particularly, we combine gating mechanism with the graph convolution operation to address the over-smoothing problem. The experimental results demonstrate that our approach significantly outperforms the baselines.

Keywords: relation extraction, graph convolutional network, chemical-induced disease, inter-sentential relation, document level

OPEN ACCESS

Edited by:
Ping Zhang,
The Ohio State University,
United States

Reviewed by:
Tong Wu,
IQVIA, United States
Hao Jiang,
National University of Defense Technology, China

***Correspondence:**
Daojian Zeng
zengdj916@163.com

Specialty section:
This article was submitted to Computational Genomics, a section of the journal Frontiers in Genetics

Received: *31 October 2020*
Accepted: *18 January 2021*
Published: *10 February 2021*

Citation:
Zeng D, Zhao C and Quan Z (2021) CID-GCN: An Effective Graph Convolutional Networks for Chemical-Induced Disease Relation Extraction. Front. Genet. 12:624307. doi: 10.3389/fgene.2021.624307

1. INTRODUCTION

Chemical-disease relation (CDR) plays an essential role in various areas of biomedical research and health care (Dogan et al., 2009). Understanding correlations between chemicals and diseases is made challenging. At present, it provides manually curated facts about CDR in the commonly used bioinformatics databases such as the Comparative Toxicogenomics Database (CTD) (Davis et al., 2017). Nevertheless, with the rapid accumulation of the biomedical literature, the manual curation not only time consuming, but also requires professional labeling staff and insufficient to keep up-to-date. Automatic extraction of CDR has attracted plenty of attention and become increasingly important.

To promote the research,the BioCreative V in 2015 proposed a new challenge for extracting CDR from the biomedical literature. The challenge included two subtasks (Wei et al., 2015): the disease named entity recognition (DNER) task and the chemical-induced disease (CID) relation extraction task. The former one is to identify diseases and chemicals from the given raw PubMed abstracts and normalize them to Medical Subject Headings (MeSH) concept identifiers. The latter one is to assess

whether there is association between chemicals and diseases denoted by MeSH identifier pairs. In this paper, we mainly focus on the CID relation extraction task.

The CID relation extraction is usually formulated as a binary classification problem. The difficulty of the task is to get a good vector representation for a pair of chemical and disease. Different from previous biomedical relation extraction tasks such as protein-protein interaction (PPI) detection and drug-drug interaction(DDI) detection, the CID relations are determined at document level, i.e., an entity is often represented in multiple mentions and the relations could be described across sentences. The main challenge of document level relation extraction is to deal with multiple entity mention pairs of the same concepts all over a document and capture inter-sentence relations when two entities are not in the same sentence.

Traditional methods handle CID relation extraction as two separated tasks (intra- and inter-sentence relation extraction) (Zhou et al., 2015; Qian and Zhou, 2016; Gu et al., 2017). The results of these two subtasks are merged through a post-processing way to obtain CID relations between entity concepts at document level. The development of such methods mainly draws on the traditional sentence level relation extraction. Feature-based methods (Qian and Zhou, 2016) and kernel-based methods (Zhou et al., 2015) have appeared one after another. With the revival of deep learning in recent years, researchers have used various neural networks, such as convolutional neural networks (Gu et al., 2017), to automatically learn features. The separated framework classifies multiple mention-level pairs, which is simple and easy to implement. However, it ignores the interactions in multiple mentions of the target entities in different sentences, which are especially useful to identify the inter-sentential relations.

In order to take full advantage of correlations among different mentions in a document, the graph-based approaches are proposed for CID relation extraction. The graph-based models interpret words as nodes and edges as intra- and inter-sentential relations between the words. Quirk and Poon (2017) build a document graph with different dependency edges. Following this work, researchers exploit graph LSTM (Peng et al., 2017), graph state LSTM (Song et al., 2018) or RNNs on dependency tree structures (Gupta et al., 2019). These methods can simultaneously capture intra- and inter-sentential features, but they do not aggregate the features of multiple mentions. Recently, many approaches are proposed to address this problem. Christopoulou et al. (2019a) is one of the most powerful systems, which use an edge-oriented graph (EoG) neural model to learn the representation of mention pairs.

Although edge-oriented models such as *EoG* have good performance, it only focuses on the edge representation of the graph and ignores the representation of the nodes in the graph. On the one hand, with the multi-hop reasoning over the document graph, the meaning of some edges on the path of multiple entity pairs will become overlapping and vague. On the other hand, since it only focuses on the representation of the node pair, it may lose the specific and related information (e.g., entity or mention type) of the node itself, which is very important in document-level relation extraction. For learning the representation of the nodes in the graph, a powerful approach is Graph Convolutional Networks (GCNs), which have achieved state-of-the-art results in various application areas on real-world datasets. The basic idea is to carry out convolution filtering on the graph and update the node representations by propagating information between nodes. Besides, by treating objects as nodes and connecting related nodes, GCNs can be adopted to various graph-based multi-hop inference tasks. However, most of the state-of-the-art GCN models are shallow due to the over-smoothing problem. The over-smoothing means that after multi-layer graph convolution, the effect of Laplacian smoothing causes node representation toward a space that contains limited distinguished information. This issue also drives the GCN difficult to model the relation between long-distance nodes, which is critical in CID relation extraction.

In this paper, we propose an effective Graph Convolutional Networks (GCNs) for CID relation extraction (CID-GCN). Similar to Christopoulou et al. (2019a), we construct a heterogeneous graph which contains mention, sentence and entity nodes. By processing all the entities and mention nodes in the document in a unified manner, the intra- and inter-sentence relation facts can be extracted simultaneously in a model. Instead of using a walk-based method, we subsequently exploit graph convolution operation to aggregate interactive information on the constructed graph. Graph convolution operation applies the same linear transformation to all the neighbors of a node followed by a non-linear activation function. In order to make the graph better adapt to CID relation extraction, we stack multiple graph convolution operations for multi-hop reasoning over the heterogeneous graph. To address the smoothing problem, we propose an enhanced gating mechanism that controls the connections between convolutional network layers. Finally, we enumerate possible entity combinations and incorporate a softmax classifier to get the relation of entity pairs. The contributions of this paper can be summarized as following:

- We propose a novel heterogeneous graph-based node-oriented model for CID relation extraction which simultaneous extract intra- and inter-sentence relation facts.
- We propose a gating mechanism for GCNs which can better capture the relation between long-distance nodes by alleviating the over-smoothing problem.
- We conduct wide experiments on a public document-level biomedical datasets. Experimental results show that the proposed method outperforms several strong baselines.

2. RELATED WORK

Relation extraction is the widely studied task of automatically retrieving structured information (relational facts) from text. It has received widespread attention as the key component for building Knowledge Graphs. According to the text of input, relation extraction falls into sentence-level and document-level methods. The CID relation extraction is a recently introduced task. From the task definition (Wei et al., 2015), CID relations

are typically determined at document level, meaning that this task should consider both intra- and inter-sentence relations.

Early studies tackle the CID relation extraction based on traditional sentence-level relation extraction methods. It is worth noting that for inter-sentence relations, multiple sentences are generally taken as a whole, and the cross-sentence features are extracted. The task is usually considered as a classification problem. Jiang et al. (2015) exploit word embeddings and linguistic features to represent the relation between chemicals and diseases. Then, they use a logistic regression model with a heuristic post-processing method to get the CID relations. Zhou et al. (2015) apply the shortest dependency path tree kernel with support vector machine (SVM) for the CID relation classification. Qian and Zhou (2016) incorporating different maximum entropy (ME) classifiers with lexical and syntactic features to extract cross-sentence relations. In addition, methods using prior knowledge and external resource have been proved to be effective. Pons et al. (2016) add prior knowledge about chemicals and diseases from a graph database. Peng et al. (2016) incorporate weakly labeled data to improve the performance. With the development of deep learning, (Zhou et al., 2016) propose a hybrid method which adopts LSTM to generate semantic representations. Gu et al. (2017) employ CNN to learn the context and dependency representations. Nguyen and Verspoor (2018) further use character-based word embeddings to improve the CNN model.

The above methods, especially the introduction of deep neural networks, have greatly promoted the development of this task. However, these methods ignore the interactions in multiple mentions of the target entities in different sentences. Recently, the graph-based approaches are proposed for the CID relation extraction. Quirk and Poon (2017) build a document graph with different dependency edges. They incorporate both standard dependencies and discourse relations and provide a unifying way to model relations intra- and inter-sentences. Peng et al. (2017) exploit graph LSTM to extract n-ary relations that span multiple sentences. Following this work, graph state LSTM (Song et al., 2018) and RNNs on dependency tree structures (Gupta et al., 2019) are used to model inter-sentence relations. These methods can simultaneously capture intra- and inter-sentential features, but they do not aggregate the features of multiple mentions. To address this shortcoming, Verga et al. (2018) form pairwise predictions over multiple sentences using a self-attention encoder, and aggregate the predictions by multi-instance learning. Christopoulou et al. (2019a) use an edge-oriented graph neural model to learn the representation of mention pairs. Nan et al. (2020) develop a refinement strategy to automatically induce the latent document-level graph, which helps to reason relations across sentences.

3. TASK DEFINITION

We follow the definition of CID relation extraction from BioCreative V community. The input of CID relation extraction task is a well-annotated biomedical document from PubMed articles. The output is a ranked list $\langle chemical, disease \rangle$ pairs with normalized concept identifiers for which chemical-induced diseases are associated in the document. We introduce a document instance from CDR dataset and shown in **Figure 1** as an example to help understand the task. As illustrated in the **Figure 1**, given a biomedical document composed with eight sentences and 12 chemical or disease entity mentions corresponding to six concept ID. In the document, entities may have multiple mentions scattered in different sentences with same color. The goal of CID task is to find CID relations in concept pairs (e.g., D003630 and D009503), rather than two mentions. In order to identify the relational fact $\langle D003630; chemical\_induced\_disease; D009503 \rangle$, *D003630* is chemical entity concept means daunorubicin and *D009503* is disease entity concept means neutropenia, one has to first identify the fact that *daunorubicin in advanced Kaposi 's sarcoma* is located in the title, then identify the facts *neutropenia* is the symptom of three subjects in the clinical trial from Sentence 5 in the abstract, and finally infer from these facts that *D003630* can induce *D009503*. Clearly, the process requires reading and reasoning over multiple sentences in the document.

Formally, the document-level chemical-induced disease (CID) relation extraction task can be formulated as follows. Given an input document d composed of T sentences $s_1, s_2, \cdots, s_T$, with N entity mentions $m_1, m_2, \cdots, m_N$, and R normalized entity concept identifiers $c_1, c_2, \cdots, c_R$. The task aims at extracting the relation r between each pair of chemical entity c_i and disease entity c_j, for $i, j = 1, 2, \cdots, R$. the relation $r = 1$ denotes that the chemical entity c_i and the disease entity c_j has the *Chemical-induced Disease* relation, $r = 0$ denotes there is no relation between two entities.

4. THE OVERVIEW OF OUR MODEL

In this paper, we present an effective graph convolutional neural network for document-level chemical-induced disease relation extraction: CID-GCN. It consists of three modules, namely (i) encoding layer, (ii) graph aggregation layer, and (iii) classifier layer. **Figure 2** illustrates the detailed structure of the model.

The purpose of the encoding layer is to learn the feature vector representation of the three nodes from the input document. The backbone of the nodes encoder network is an RNN. RNN is widely used for learning sequential and time-dependent structures inherent in the text, and achieved state-of-the-art results in many Natural language processing (NLP) tasks of high-value biomedical domain in recent years, including biomedical Named Entity Recognition (NER), biomedical QA etc. In order to construct a document-level graph, we encode three different types of nodes, respectively mention nodes, entity nodes, and sentence nodes. Specifically, given a document as the model input, it first generates a deep contextualized representation for each sentence using RNN with LSTM cell. Next, it constructs the representation of nodes based on the

TITLE: Liposomal daunorubicin in advanced Kaposi 's sarcoma
ABSTRACT: We report a non-randomized Phase II clinical trial to assess the efficacy and safety of liposomal daunorubicin in the treatment of AIDS related Kaposi 's sarcoma.
Eleven homosexual men with advanced Kaposi 's sarcoma were entered in the trial .
A partial response was achieved in four , of whom two subsequently relapsed .
Stabilization of Kaposi 's sarcoma occurred in the remaining six , maintained until the end of the trial period in four .
The main problem encountered was haematological toxicity , with three subjects experiencing severe neutropenia.
There was no evidence of cardiotoxicity.
In this small patient sample , liposomal daunorubicin was an effective and well tolerated agent in the treatment of Kaposi 's sarcoma.

Entity Mention		Entity Type	Concept ID
m1	daunorubicin	Chemical	D003630
m2	Kaposi 's sarcoma	Disease	D012514
m3	daunorubicin	Chemical	D003630
m4	AIDS	Disease	D000163
m5	Kaposi 's sarcoma	Disease	D012514
m6	Kaposi 's sarcoma	Disease	D012514
m7	Kaposi 's sarcoma	Disease	D012514
m8	toxicity	Disease	D064420
m9	neutropenia	Disease	D009503
m10	cardiotoxicity	Disease	D066126
m11	daunorubicin	Chemical	D003630
m12	Kaposi 's sarcoma	Disease	D012514

Relation		Relation Type
r1	D003630(daunorubicin) | D012514(Kaposi 's sarcoma)	non chemical-induced-disease
r2	D003630(daunorubicin) | D000163(AIDS)	non chemical-induced-disease
r3	D003630(daunorubicin) | D064420(toxicity)	non chemical-induced-disease
r4	D003630(daunorubicin) | D009503(neutropenia)	chemical-induced-disease
r5	D003630(daunorubicin) | D066126(cardiotoxicity)	non chemical-induced-disease

FIGURE 1 | Introduction to document-level chemical-induced disease relation extraction.

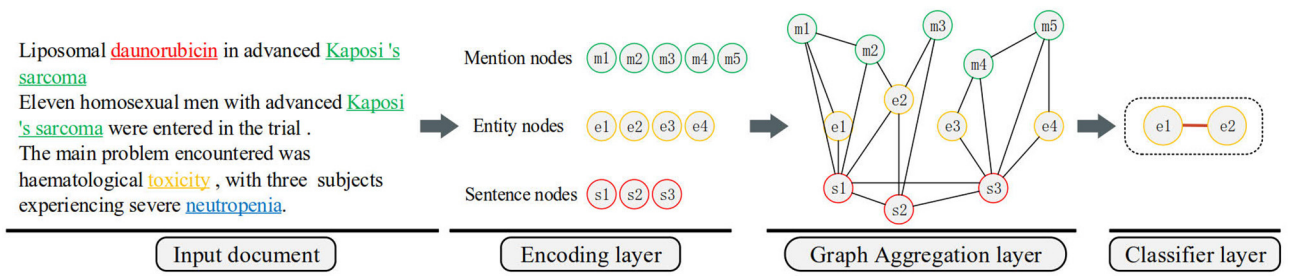

FIGURE 2 | The overall architecture of the proposed neural model.

sentence representations. The specific details are explained in section 5.

The graph aggregation layer is devised to inference entity nodes interactions in the document. First, we construct a graph by connecting the graph nodes based on the natural associations among the three nodes in the document. After building the heterogeneous document-level graph, we utilize GCN to encode entity nodes by exploiting other interactions in the document. Recently, it has demonstrated that GCNs is powerful in processing relational reasoning on graphs. We stack multiple graph convolutional operations over the graph to map the node vectors into a set of new node representations. We will provide a brief recap of GCN and introduce its application in section 6.

Last, the output classifier module gives the relation prediction from two graph representations of entity nodes. In section 7, the loss function and the training process will be described further.

5. ENCODING LAYER

As we mentioned in section 4, encoding layer aims to learn the features related to specific three nodes.

5.1. Word Embeddings

Word embeddings are learned from a large amount of unlabeled data and have been shown to be able to capture the meaningful semantic regularities of words (Bengio et al., 2003; Erhan et al., 2010). The input tokens of the neural network model are

a sequence of discrete variables. We usually transform these discrete variables into vector representations in the NLP area.

In this work, given a document d, we use NLTK python library[1] to convert the corresponding multiple sentences $s_1, s_2, \cdots, s_T$ into multiple sequences of token, respectively. Then, we use the pre-trained word embeddings trained by two corpora: PubMed Central Open Access subset (PMC) and PubMed (Chiu et al., 2016). All the input word tokens will be transformed into low-dimensional vectors by looking up word embeddings tables, respectively, in sentence units. In this work, we denote the dimension of word embeddings by d_w. These word embeddings of each sentence are transformed for the subsequent layers.

5.2. LSTM Encoding Layer

RNN has been widely exploited to deal with variable-length sequence input and successfully applied in various NLP tasks. The long-distance history is stored in a recurrent hidden vector which is dependent on the immediate previous hidden vector. LSTM (Hochreiter and Schmidhuber, 1997) is one of the popular variations of RNN to mitigate the gradient vanish problem of RNN. LSTM has three gates(input i, forget f and output o), and a cell memory vector c. The input gate can determine how incoming vectors x(t) alter the state of the memory cell. The output gate can allow the memory cell to have an effect on the outputs. Finally, the forget gate allows the cell to remember or forget its previous state. Given an input sequence $\mathbf{x} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n\}$, where $\mathbf{x}_1$ is a d_w dimension vector. The hidden vector $\mathbf{h}_t$ (the dimension is d_h) at the time step $t(1 \leq t \leq n)$ is calculated as follows:

$$\begin{aligned}
\mathbf{i}_t &= \sigma(\mathbf{W}_i\mathbf{x}_t + \mathbf{U_i}\mathbf{h}_{t-1} + b_i) \\
\mathbf{f}_t &= \sigma(\mathbf{W}_f\mathbf{x}_t + \mathbf{U}_f\mathbf{h}_{t-1} + b_f) \\
\mathbf{o}_t &= \sigma(\mathbf{W}_o\mathbf{x}_t + \mathbf{U}_o\mathbf{h}_{t-1} + b_o) \\
\widetilde{\mathbf{C}}_t &= tanh(\mathbf{W_c}x_t + \mathbf{U_c}h_{t-1} + b_c) \\
\mathbf{C}_t &= \mathbf{i}_t \odot \widetilde{\mathbf{C}}_t + \mathbf{f}_t \odot \mathbf{C}_{t-1} \\
\mathbf{h}_t &= o_t \odot tanh(\mathbf{C}_t)
\end{aligned} \tag{1}$$

where $W_i, W_f, W_o, W_c \in \mathbb{R}^{d\times h}$ and $U_i, U_f, U_o, U_c \in \mathbb{R}^{h\times h}$ are weight parameters and $b_i, b_f, b_o, b_c \in \mathbb{R}^h$ are bias parameters, and $\odot$ denotes element-wise multiplication. The subscripts i, f, o represent input gate, forget gate and output gate, respectively.

In this work, given that a document contains T sentences $s_1, s_2, \cdots, s_T$, we do the following operations for each sentence. First, we use Bidirectional LSTM (BiLSTM) as the sentence encoder to read the input sequence in both left-to-right and reverse order. Second, we combine bidirectional information for each word by averaging the forward and the backward output. Thus, given a sentence $\mathbf{s_i} = \{x_1, x_2, \cdots, x_n\}$ as a sequence of tokens, the LSTM encoding layer is responsible to map each token to the continuous embedding representations as $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_n\}$. After encoding contextualized representations of all the sentences, There are three types of node that need to be constructed in CID-GCN.

[1]https://www.nltk.org

5.2.1. Mention Node Representation

The mention node is intended to represent different mentions of entities that appear in the document. A mention node is represented by the average of the hidden vectors of all words contained in the mention after LSTM encoding. Assuming that a document has N mentions, the representation of mention nodes is formed as $N_{m_j} = [\mathbf{avg}_{h_i \in m_j}(h_i); t_m]j = 1, 2, \cdots, N$. where t_m is a node type embedding for mention.

5.2.2. Entity Node Representation

Similar to the structure of the mention node, the structure of the entity node is represented by the average of the representations of all the mentioned nodes corresponding to the entity. Assuming that a document has R entities, the representation of entity nodes is formed as $N_{e_j} = [\mathbf{avg}_{m_i \in e_j}(m_i); t_e]j = 1, 2, \cdots, R$. where t_e is an node type embedding for entity.

5.2.3. Sentence Node Representation

A sentence node n_{s_j} is represented by the average of output at all times in $\mathbf{H_j}$: Assuming that a document has T sentences, the representation of sentence nodes is formed as $N_{s_j} = [\mathbf{avg}_{h_i \in s_j}(h_i); t_s]j = 1, 2, \cdots, T$. where t_e is an node type embedding for sentence.

6. GRAPH AGGREGATION LAYER

GCN is a powerful approach for mining the structural features on the graph. We use the GCN to capture the correlations of multiple entity nodes. In this section, we will introduce the structure of document graph and describe the preliminary and detail of the GCN.

6.1. Document Graph Construction

A graph is made up of nodes (also called vertices) which are connected by edges (also called links). Normally, it is an ordered pair $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is a set of nodes and $\mathcal{E}$ is a set of edges. We can mathematically represent a graph with n nodes by an adjacency matrix $\mathbf{A} \in \mathbb{R}^{n\times n}$, where $\mathbf{A}_{ij} = 1$ if an edge exists between node i and j, otherwise 0.

In this paper, we construct a document-level heterogeneous graph by connecting the three nodes constructed in the section above. Specifically, we first constructed a total of $N + R + T$ nodes of mention, entity, and sentence. Next, we connect the graph nodes based on the natural associations between the three nodes in the document to obtain the adjacency matrix $\mathbf{A} \in \mathbb{R}^{(N+R+T)\times(N+R+T)}$ of the document graph. These natural connections have the following 5 situations:

- Mention-to-Mention. If there are two mentions appear in the same sentence, there is an implicit meaning that cannot be ignored. These pair of two mention nodes should be connected.
- Sentence-to-Sentence. We connect every sentence nodes to model global information.
- Mention-to-Sentence. Since every mention must appear in a sentence, we connect all mentions to the sentence node where they are located.

- Mention-to-Entity. Similar to Mention-to-Sentence, we connect all the mentioned nodes to their corresponding entity node.
- Entity-to-Sentence. To model the diversity of entities in the document, we connect all entities to the sentence node where their mentions have appeared.

6.2. Graph Convolutional Networks

GCN is an extension of convolutional neural network and can be operated to encode graphs. It carry out convolution filtering on the graph and update the node representations by propagating information between nodes. We stack all the three types of constructed nodes into the node set $\mathcal{V}$. The set of nodes $\mathcal{V}$ are represented as d dimensional vectors $\mathbf{V} \in \mathbb{R}^{(N+R+T)\times d}$, the GCN layer on a graph can be written as a non-linear function $f(\mathbf{V}, \mathbf{A})$. When considering the stacking of multiple GCN layer and after exploiting the convolutional operation proposed in Kipf and Welling (2017), the GCN can be represented as follows:

$$\mathbf{V}^{l+1} = f(\mathbf{V}^l, \mathbf{A}) = \delta(\mathbf{A}\mathbf{V}^l\mathbf{W}^l) \tag{2}$$

where $\delta(\cdot)$ denotes an activation function, which is chosen as LeakyReLU in our experiments. As a general rule, the superscript l indicates the layer number. $\mathbf{W}^l \in \mathbb{R}^{d\times d}$ is the learnable parameters of the convolutional filter. Different graph convolution layers have different convolutional filters, which are numbered using superscript l. It is not difficult to see that multiplying the adjacency matrix is equivalent to adding the feature of its neighbor node to each node. For instance, if entity node 5 has two adjacent nodes: sentence node 1 and mention node 4, the Equation (2) can be represented in another way as:

$$V_5^{l+1} = \delta(a_{35}v_3^l\mathbf{W}^l + a_{45}V_4^l\mathbf{W}^l + a_{55}V_5^l\mathbf{W}^l) \tag{3}$$

where a_{ij} is the element at row i and column j of the adjacent matrix $\mathbf{A}$, V_i^l denotes the node representation corresponding to the i-th node of the l-th layer. In this case, the GCN aggregates all adjacent node information with the same convolution weights, and after that, the result is passed through one activation function to yield the updated node feature. In this way, the adjacent nodes in the graph affect each other, and the relation among entity nodes is learned after multiple layers of convolution operations.

Facts (Kipf and Welling, 2017; Li et al., 2018) have proved that the graph convolution is a special form of Laplacian smoothing, which mixes the features of the nodes and its neighbors.The smoothing operation makes the features of the nodes in the same cluster similar, thereby optimizing the classification task, which is the key reason why GCNs works so well. However, this also brings the potential problem of over-smoothing when stacking multiple GCN layers. The over-smoothing problems can lead to similar node representations, thus losing the discrimination of the node in the classification function. Moreover, this problem also limits the long-distance relation modeling ability of the model. However, long-distance reasoning paths are very common in the document graph we construct, because the relation between entity nodes may need to be inferred from multiple mention nodes and sentence nodes.

In order to alleviate the above problems, we propose a gating mechanism for GCNs. This mechanism divides the traditional graph convolutional layer into two steps. The first step, using the structure information of the graph to aggregate the information of adjacent nodes to passing messages on the graph, which is consistent with the operation of the traditional GCNs. The second step, using a gating mechanism to control the updating of node representations. The gating mechanism can be calculated as follows:

$$\begin{aligned} \mathbf{g}_l &= sigmoid(\mathbf{W}_g\mathbf{V}_{l+1} + \mathbf{U}_\mathbf{g}\mathbf{V}_l + b_g) \\ \mathbf{V}_{l+1} &= \mathbf{V}_{l+1} \odot \mathbf{g}_l + \mathbf{V}_l \odot (1 - \mathbf{g}_l) \end{aligned} \tag{4}$$

where $\mathbf{W}_g \in \mathbb{R}^{d\times d}$ and $\mathbf{U}_g \in \mathbb{R}^{d\times d}$ are two learnable parameters. The gate $\mathbf{g}_l$ controls the new node representation $\mathbf{V}_{l+1}$ update of each layer by considering the node representations generated by the previous layer $\mathbf{V}_l$ and the current graph convolutional layer $\mathbf{V}_{l+1}$. The gating mechanism aims to save the distinguished local information belonging to the current node representations itself after each graph convolution operation. Combined with effective global information and unique local information, the model can better understand the document graph and learn more distinguishable node representations, thereby alleviating the over-smoothing problem caused by multi-layer GCNs. Besides, compared with the edge update mechanism of walk-base method (Christopoulou et al., 2019b), our proposed gating mechanism does not require manual tuning of hyperparameters to determine the contribution of each hop.

7. CLASSIFIER LAYER

After multiple times of aggregation, we obtain a set of new representations of all the nodes. For each entity chemical-disease pair (ei,ej), we use a bilinear function to compute the probability for chemical-induced disease relation as:

$$\mathbf{P}(\mathbf{r}|\mathbf{e_i}, \mathbf{e_j}) = softmax(\mathbf{e_i}^T\mathbf{W}_{cls}\mathbf{e_j}) \tag{5}$$

where $\mathbf{W}_{cls} \in \mathbb{R}^{d\times k\times d}$ is a learnable parameter matrix. k is the number of labels, which is 2 in this work cause CID relation extraction is a binary classification problem.

In this paper, we use the stochastic gradient descent (SGD) algorithm to minimize the log likelihood function. The loss function of our model is:

$$\mathcal{L} = -\sum_{d\in\mathcal{T}} \log p(r_{e_i,e_j} = r^*_{e_i,e_j}|d) \tag{6}$$

where $\mathcal{T}$ represents the training set, $r^*_{e_i,e_j}$ is the gold label for the relation between entity chemical-disease pair (ei,ej) in document d. During training, we minimize the loss function $\mathcal{L}$ of the gold CID relations.

8. EXPERIMENTS

8.1. Datasets and Setting

We evaluate the performance of our model on **CDR** dataset proposed by BioCreative V. This dataset contains a total of 1,500

TABLE 1 | Main results on CDR datasets.

Model	Description	Precision	Recall	F1
Zhou et al. (2015)	*CNN*	41.1	55.3	47.2
Zhou et al. (2016)	*LSTM+SVM*	64.9	49.3	56.0
Gu et al. (2017)	*CNN+Inter_ME+PP*	55.7	68.1	61.3
Nguyen and Verspoor (2018)	*CNN+Char*	57.0	68.6	62.3
Sahu et al. (2019)	*GCNN*	52.8	66.0	58.6
Peng et al. (2017)	*Graph LSTM*	62.1	64.2	63.1
Verga et al. (2018)	*BRAN*	55.6	70.8	62.1
Wang et al. (2020)	*GCN+Multihead Attention*	56.3	**72.7**	63.5
Christopoulou et al. (2019a)	*EoG*	62.1	65.2	63.6
Nan et al. (2020)	*LSR*	-	-	64.8
Our model	*CID-GCN*	**64.2**	66.4	**65.3**

Bold marks highest number among all models.

TABLE 2 | Performance of EoG and CID-GCN with different pre-trained word embeddings.

Model	F1 (%)
EoG *(random)*	61.41
EoG *(GloVe)*	63.01
EoG *(PubMed)*	63.62
CID-GCN *(random)*	62.55
CID-GCN *(GloVe)*	64.46
CID-GCN *(PubMed)*	**65.32**

Bold marks the highest number among all models.

PubMed articles, 500 articles each for the training, development and test set. Each articles is manually annotated chemical mentions and disease mentions, the MeSH identifiers of chemical entity and disease entity, and the CID relation between the chemical entities and disease entities. **Table 1** details the diseases and related annotations of these three data sets.

In our experiments, all hyper-parameters are tuned through cross validation on training set and development set. We initialize network with pre-trained embedding with a dimension of 300.The hidden state of one-side LSTM is 300. The sizes of the three types of node embedding is 50.The number of layers of the GCNs is 4. The experiments are trained with an NVIDIA RTX 2080Ti GPU. It took about 10 min per epoch.

8.2. Baselines and Evaluation Metrics

To evaluate the performance of the proposed method, we compare our model with six competitive baselines, as follows: (1) **CNN** (Zhou et al., 2015). (2) **LSTM+SVM** (Zhou et al., 2016). (3) **CNN+Inter_ME+PP** (Gu et al., 2017). (4) **CNN+Char** (Nguyen and Verspoor, 2018). (5) **GCNN** (Sahu et al., 2019). (6) **Graph LSTM** (Peng et al., 2017). (7) **BRAN** (Verga et al., 2018). (8) **GCN+Multihead Attention** (Wang et al., 2020). (9) **EoG** (Christopoulou et al., 2019a). (10) **LSR** (Nan et al., 2020). We use precision, recall, and F1 score to evaluate the performance.

8.3. Main Results

To evaluate the performance of the proposed method, we first compare our model with the baseline methods. The results are shown in **Table 1**, from which we can observe that:

(1) Compared with the current graph-based model, our model has achieved the best results. In detail, compared with *Graph LSTM* and *BRAN*, the improvements of our model are 2.2 and 3.2% in F1, respectively. It indicates that our method can better take advantage of the rich correlations among entities at document level. Furthermore, compared with *GCNN* and *GCN+Multihead attention* which both use plain-GCN, the improvements of our model are 6.7 and 1.8% in F1 score, respectively. This is due to our reasonable method for document graph construction. Similar to *EoG*, we construct a heterogeneous graph which contains mention, sentence and entity nodes. However, our model outperforms *EoG* with F1 score of 1.7%, and 2.1% in precision 1.2% in recall. The main reason is that the graph aggregation layer of our model can encode more entity-relation information for relation classification. The gating mechanism proposed in our model enables the aggregation layer to encode the complete graph structure without losing the information when modeling the information of multi-hop nodes. To further compare EoG with CID-GCN, we analyzed the performance of EoG and CID-GCN using different word embeddings. As shown in **Table 2**, CID-GCN is superior to EoG in random initialization *(random)*, general domain *(GloVe)* (Pennington et al., 2014), domain-specific *(PubMed)* (Chiu et al., 2016) word embeddings. *LSR* also exploits graph convolution operation on the document graph but uses the dense connection to address the over-smoothing problem, our model outperforms it with F1 score of 0.5%. Thus, Our method can make better capture the correlation between chemical entities and disease entities in the CDR dataset.

(2) Compared with the current non-graph-based model, our model also has achieved the best performance in CID relation extraction task. In fact, these methods do not perform as well as graph-based methods on the CDR dataset. Specifically, *CNN* is characterized by a low precision, which is caused by its deficiency in cross-sentence relation facts. The other three methods utilize different inter- and intra-sentence models and merge the final predictions. In the detailed analysis of the datasets, we find that 30% CID relations in the test set belong to entity pairs that cross sentences. Compared with *LSTM+SVM*, the improvements of our model is 9.3% in F1. In contrast, the performance gap of other baselines is relatively small, 4.0% and 3.0% in F1, respectively.

(3) In order to verify the superiority of our model in extracting cross-sentence CID relations, we separately verified the performance of the model in inter- and intra-sentence. We selected two comparison models from graph-based and non-graph-based models, *CNN+Inter_ME+PP* and *EoG*, respectively.

TABLE 3 | Experimental results in intra- and inter-sentence CID relations.

Model	Intra (%)			Inter (%)		
	Precision	Recall	F1	Precision	Recall	F1
CNN+Inter_ME+PP	59.7	55.0	57.2	51.9	7.0	11.7
EoG	64.0	73.0	68.2	56.0	**46.7**	50.9
Our model	**68.1**	**75.9**	**71.8**	**62.2**	45.2	**52.4**

Bold marks the highest number among all models.

TABLE 4 | Ablation study for the graph aggregation layer, where "w/o" indicates without and "w/" indicates with.

Model	Precision	Recall	F1	Intra-F1	Inter-F1
Our model	**64.2**	**66.4**	**65.3**	**71.8**	**52.4**
w fully-connected	60.6	58.8	59.7	68.5	43.8
w/o gating mechanism	61.2	52.3	56.4	62.9	44.6
w/o aggregation layer	53.0	51.0	53.1	60.8	30.6

Bold marks the highest number among all models.

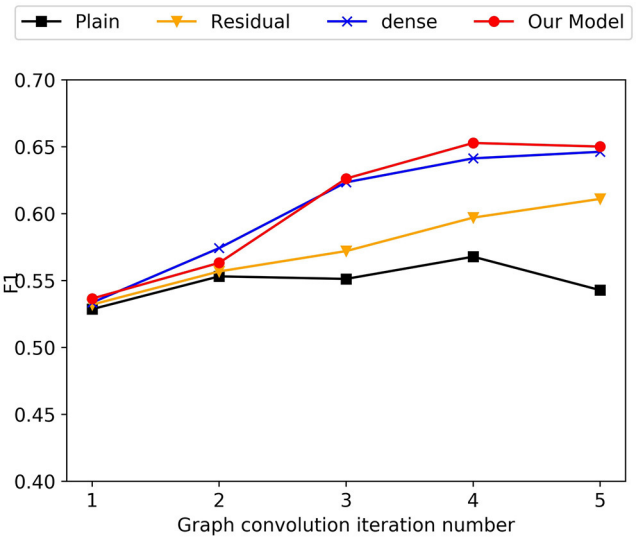

FIGURE 3 | Experimental results of different GCN connection mechanisms.

Table 3 depicts the results of our proposed model, in comparison with the two baseline selected above. As it can be observed, *CNN+Inter_ME+PP* obtained a very low F1 score when recognizing the inter-sentence CID relation. The reason is it cannot well capture the interactions in multiple mentions of the target entities in different sentences. Compared with *EoG* and *BRAN*, the improvements of our model is also considerable.

Our model not only can simultaneous extract intra- and inter-sentence relation facts, but also capture better the interactions between entities regardless of whether they cross sentences. We further verify the advantages of the graph aggregation layer with different connect mechanism for GCNs in the following subsection. Therefore, our model outperforms all the baselines and more suited to the CID relation extraction task.

8.4. Ablation Experiments

To investigate the effectiveness of the graph aggregation layer proposed in CID-GCN, we conduct an ablation study using the test set of CDR dataset. **Table 4** shows the results of ablation study. From the table, we find that:

(1) When we change the document graph to a fully connected graph, even with the existence of the gating mechanism, the performance of the model still drops rapidly. This results show the importance of a reasonable document graph structure.

(2) Both intra-F1 and inter-F1 scores drop when we remove the gating mechanism. This results confirm the existence of the over-smoothing problem, and also show that the gating mechanism does enable the model to better capture the relation between long-distance nodes.

(3) Finally, when we remove the aggregation layer, inter-F1 scores drops dramatically. According to our statistics on the CDR dataset, there are more than 54% entities have multiple mentions in different sentences. This results prove our proposed graph aggregation layer can effectively extract the inter-sentence CID relation facts by reading and reasoning over multiple sentences in the document graph we construct.

8.5. Analysis Using GCN

Recently, GCNs have shown excellent performance in various NLP tasks. Though effective, most of the current GCN models are shallow due to the smoothing problem. As illustrated in the **Figure 3**, our model also suffers from over-smoothing problems when using plain-GCN. The plain-GCN achieve their best performance with 4-layer models, but the performance is still poor. In the heterogeneous graphs we construct, we give various nodes different meanings, which may aggravate the over-smoothing problem.

To alleviate this problem, we tried three different connection methods. (1) Similar to ResNet, we add residual connections between different layers of GCNs. (2) Similar to DenseNet, we concatenates the outputs of all graph convolution layers to get the final mention, sentence and entity node representations. (3) Inspired from the forget gate in LSTM, we propose a gating mechanism for GCNs.

We conduct experiments to investigate the effectiveness of the three enhanced connect mechanisms on different layers. As shown in the **Figure 3**, the three enhanced connect mechanisms slows down the over-smoothing problem to varying degrees. Among them, our proposed gating mechanism achieves the best performance on 4-layer GCNs. It means the gating mechanism we proposed is more suited to the CID relation extraction task and the document graph we construct.

8.6. Case Study

In order to investigate the advantages of our model, we conduct case study on the test set of CDR. Compared to all the baselines, our model can extract more inter-sentence CID relations correctly. For example, in the document number 57355

entitled "Long-term propranolol therapy in pregnancy : maternal and fetal outcome." The main idea of this article is to study the relationship between 6 diseases and long-term propranolol treatment during pregnancy through two sets of experiments. There are 9 sentences in this document, and the first 8 sentences contain a mention of the same chemical entity "propranolol (D011433)." The experiment of this document uses exclusion division to exclude the first 5 diseases. The last sentence "Growth retardation, however, appears to be significant in both of our series" finally pointed out the CID relation between chemical entity "propranolol (D011433)" and disease entity "Growth retardation (D005317)" and it do not contain "propranolol (D011433)" entity. As can be seen from this case, our model can correctly extract the inter-sentence CID relations by reading and reasoning over multiple sentences in the document.

9. CONCLUSION

In this paper, we propose CID-GCN, an effective Graph Convolutional Networks with gating mechanism, for CID relation extraction. First, we construct a heterogeneous graph which contains mention, sentence and entity nodes and connect the nodes based on the natural associations among the three nodes in the document. In particular, in order to solve the over-smooth problem of graph convolutional neural networks on heterogeneous graphs, we propose a gating mechanism to connect different GCN layers. Experimental results on CDR datasets indicate that our proposed model is effective and outperforms several strong baseline methods.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found here: http://ctdbase.org/.

AUTHOR CONTRIBUTIONS

DZ: conceptualization and methodology. CZ: writing the original draft and experiments. ZQ: investigation and visualization. All authors have read and approved the final manuscript.

FUNDING

This work was supported by the Hunan Provincial Natural Science Foundation of China (Grant Nos. 2020JJ4624 and 2019JJ50655), the Scientific Research Fund of Hunan Provincial Education Department (Grant No. 19A020).

REFERENCES

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.* 3, 1137–1155. doi: 10.1007/3-540-33486-6_6

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016). "How to train good word embeddings for biomedical NLP," in *Proceedings of the 15th Workshop on Biomedical Natural Language Processing* (Berlin), 166–174. doi: 10.18653/v1/W16-2922

Christopoulou, F., Miwa, M., and Ananiadou, S. (2019a). "Connecting the dots: document-level neural relation extraction with edge-oriented graphs," in *Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)* (Hong Kong: Association for Computational Linguistics), 4925–4936. doi: 10.18653/v1/D19-1498

Christopoulou, F., Miwa, M., and Ananiadou, S. (2019b). A walk-based model on entity graphs for relation extraction. *arXiv[Preprint].arXiv:1902.07023*. doi: 10.18653/v1/P18-2014

Davis, A. P., Grondin, C. J., Johnson, R. J., Sciaky, D., King, B. L., McMorran, R., et al. (2017). The comparative toxicogenomics database: update 2017. *Nucleic Acids Res*. 45, D972–D978. doi: 10.1093/nar/gkw838

Dogan, R. I., Murray, G. C., Névéol, A., and Lu, Z. (2009). Understanding pubmed? User search behavior through log analysis. *Database* 2009. doi: 10.1093/database/bap018

Erhan, D., Bengio, Y., Courville, A., Manzagol, P.-A., Vincent, P., and Bengio, S. (2010). Why does unsupervised pre-training help deep learning? *J. Mach. Learn. Res.* 11, 625–660. doi: 10.5555/1756006.1756025

Gu, J., Sun, F., Qian, L., and Zhou, G. (2017). Chemical-induced disease relation extraction via convolutional neural network. *Database* 2017. doi: 10.1093/database/bax024

Gupta, P., Rajaram, S., Schütze, H., Andrassy, B., and Runkler, T. (2019). Neural relation extraction within and across sentence boundaries. *arXiv preprint arXiv:1810.05102*. doi: 10.1609/aaai.v33i01.33016513

Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Comput*. 9, 1735–1780. doi: 10.1162/neco.1997.9.8.1735

Jiang, Z., Jin, L., Li, L., Qin, M., Qu, C., Zheng, J., et al. (2015). "A CRD-WEL system for chemical-disease relations extraction," in *The Fifth BioCreative Challenge Evaluation Workshop* (Sevilla), 317–326.

Kipf, T. N., and Welling, M. (2017). "Semi-supervised classification with graph convolutional networks," in *Proceedings of the 5th International Conference on Learning Representations, ICLR '17* (Toulon).

Li, Q., Han, Z., and Wu, X.-M. (2018). Deeper insights into graph convolutional networks for semi-supervised learning. *arXiv[Preprint].arXiv:1801.07606*. Available online at: https://ojs.aaai.org/index.php/AAAI/article/view/11604

Nan, G., Guo, Z., Sekulic, I., and Lu, W. (2020). "Reasoning with latent structure refinement for document-level relation extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (Seattle: Association for Computational Linguistics), 1546–1557. doi: 10.18653/v1/2020.acl-main.141

Nguyen, D. Q., and Verspoor, K. (2018). "Convolutional neural networks for chemical-disease relation extraction are improved with character-based word embeddings," in *Proceedings of the BioNLP 2018 Workshop* (Melbourne, VIC: Association for Computational Linguistics), 129–136. doi: 10.18653/v1/W18-2314

Peng, N., Poon, H., Quirk, C., Toutanova, K., and Yih, W. T. (2017). Cross-sentence n-ary relation extraction with graph LSTMs. *Trans. Assoc. Comput. Linguist.* 5, 101–115. doi: 10.1162/tacl_a_00049

Peng, Y., Wei, C.-H., and Lu, Z. (2016). Improving chemical disease relation extraction with rich features and weakly labeled data. *J. Cheminform*. 8:53. doi: 10.1186/s13321-016-0165-z

Pennington, J., Socher, R., and Manning, C. D. (2014). "Glove: global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (Doha), 1532–1543. doi: 10.3115/v1/D14-1162

Pons, E., Becker, B., Akhondi, S., Afzal, Z., van Mulligen, E. M., and Kors, J. (2016). Extraction of chemical-induced diseases using prior knowledge and textual information. *Database* 2016. doi: 10.1093/database/baw046

Qian, L., and Zhou, G. (2016). Chemical-induced disease relation extraction with various linguistic features. *Database* 2016:baw042. doi: 10.1093/database/baw042

Quirk, C., and Poon, H. (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* (Valencia: Association for Computational Linguistics), 1171–1182. doi: 10.18653/v1/E17-1110

Sahu, S. K., Christopoulou, F., Miwa, M., and Ananiadou, S. (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* (Florence: Association for Computational Linguistics), 4309–4316. doi: 10.18653/v1/P19-1423

Song, L., Zhang, Y., Wang, Z., and Gildea, D. (2018). "N-ary relation extraction using graph-state LSTM," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing* (Brussels: Association for Computational Linguistics), 2226–2235. doi: 10.18653/v1/D18-1246

Verga, P., Strubell, E., and McCallum, A. (2018). "Simultaneously self-attending to all mentions for full-abstract biological relation extraction," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (New Orleans: Association for Computational Linguistics), 872–884. doi: 10.18653/v1/N18-1080

Wang, J., Chen, X., Zhang, Y., Zhang, Y., Wen, J., Lin, H., et al. (2020). Document-level biomedical relation extraction using graph convolutional network and multihead attention: algorithm development and validation. *JMIR Med. Inform.* 8:e17638. doi: 10.2196/17638

Wei, C. H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., et al. (2015). "Overview of the BioCreative V chemical disease relation (CDR) task," in *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop*, Vol. 14 (Sevilla).

Zhou, H., Deng, H., Chen, L., Yang, Y., Jia, C., and Huang, D. (2016). Exploiting syntactic and semantics information for chemical-disease relation extraction. *Database* 2016. doi: 10.1093/database/baw048

Zhou, H. W., Deng, H. J., and He, J. (2015). "Chemical-disease relations extraction based on the shortest dependency path tree," in *Proceedings of the Fifth BioCreative Challenge Evaluation Workshop* (Sevilla), 214–219.

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.