Expert Systems with Applications 159 (2020) 113538

Contents lists available at ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

AGCN: Attention-based graph convolutional networks for drug-drug interaction extraction

Chanhee Park, Jinuk Park, Sanghyun Park *

Department of Computer Science, Yonsei University, 50 Yonsei-ro, Seodaemun-Gu, Seoul 120-749, Republic of Korea

ARTICLE INFO

Article history:
Received 19 November 2019
Revised 5 May 2020
Accepted 6 May 2020
Available online 11 May 2020

Keywords:
Text mining
Relation extraction
Drug-drug interaction
Graph convolutional network
Attention mechanism

ABSTRACT

Extracting drug-drug interaction (DDI) relations is one of the most typical tasks in the field of biomedical relation extraction. Automatic DDI extraction from the biomedical corpus is central to the mining of knowledge hidden in the biomedical literature. Existing approaches for DDI extraction primarily focus on either the contextual or the structural information of the sentence, despite their complementary role. Also, previous studies do not even exploit the entire knowledge of the input sentence, which could lead to a loss of crucial clues. In this paper, we propose an Attention-based Graph Convolutional Networks (AGCN) to address these issues. In contrast to the existing DDI extraction methods, the AGCN is designed to leverage contextual and structural knowledge together, where GCN is employed in combination with encoders based on recurrent networks. Additionally, we apply a novel attention-based pruning strategy to optimally use syntactic information while ignoring irrelevant information, in contrast to previous rule-based pruning methods. Therefore, AGCN can take advantage of the context and structure of the input sentence as efficiently as possible. We evaluate our model using a dominant DDI extraction corpus. The experimental results demonstrate the effectiveness of our model, which outperforms existing approaches.

1. Introduction

Drug-drug interaction (DDI) extraction—one of the most typical tasks in biomedical relation extraction—aims to extract the interactions among two or more drug entities from the biomedical literature. DDI may occur when drugs are co-administered. It can increase or reduce the effects of the combined drugs and can be harmful to the human body. To prevent severe adverse drug reactions (ADRs) in advance, several databases, such as DrugBank (Wishart et al., 2018) and Drugs.com, have been created by domain experts. However, in spite of the rapid growth of biomedical literature, the majority of information is still buried in articles. Also, it is time-consuming and expensive to manually collect DDI information from natural language. Therefore, the development of a system to automatically extract DDI information from the biomedical literature has become particularly significant, especially with respect to efficiency.

Traditional relation extraction approaches based on machine learning have predominantly employed the method of feature representation or kernel design. Feature-based approaches usually exploit a diverse set of features and feed them into classifiers such as support vector machines (SVM) (Giuliano, Lavelli, Pighin, & Romano, 2007; Kim, Liu, Yeganova, & Wilbur, 2015). Thus, it is necessary to select a suitable feature set acquired through experience. Kernel-based approaches leverage syntactic information to measure the similarity between training and test sets without explicit feature representations (Airola et al., 2008; Giuliano, Lavelli, & Romano, 2006). This method also utilizes suitable kernel functions requiring careful crafting. Consequently, the performance of traditional machine learning-based methods depends on the chosen feature set or the designed kernel function.

In recent years, with the emergence of deep learning, neural network-based methods for automatic feature representation have become highly influential in DDI extraction tasks. Existing neural network-based methods can be roughly classified into two categories: sequence-based and dependency-based. Sequence-based models encode the sentence sequences into the contextualized latent features with recurrent neural networks (RNN) or convolutional neural networks (CNN) (Huang, Jiang, Zou, & Li, 2017; Liu, Tang, Chen, & Wang, 2016a). Dependency-based models incorporate dependency trees of the given sentences into neural models; this approach has proven to be very effective for relation extraction (Xu, Feng, Huang, & Zhao, 2015). Because the dependency trees involve rich structural information, dependency-based models

* Corresponding author.
E-mail addresses: channy_12@yonsei.ac.kr (C. Park), parkju536@yonsei.ac.kr (J. Park), sanghyun@yonsei.ac.kr (S. Park).

https://doi.org/10.1016/j.eswa.2020.113538
0957-4174/© 2020 Elsevier Ltd. All rights reserved.

can capture long-range syntactic relations that are ambiguous on the surface, particularly for sentences with complex or long clauses.

Although a considerable amount of research has been performed on DDI extraction, there are still a few challenging issues to be addressed to improve performance. First, previous studies have mostly focused on only one of the contextual or structural information of the sentence (Zhao, Yang, Luo, Lin, & Wang, 2016; Zhou, Miao, & He, 2018). These approaches would lack the other type of information. For instance, the dependency-based models would not be sufficient for representing the semantic meaning of the sentence, and vice versa. Additionally, existing dependency-based neural architectures (e.g., Tree-LSTM) are usually inefficient because of the difficulty of parallelism, and the models constrain the inputs to a tree structure. Second, most of the existing models do not exploit the entire knowledge of the input sentence, which could cause a major loss of information. In sequence-based models, the length of the input sentence is practically limited to the chosen maximum length. In dependency-based models, rule-based pruning strategies, such as the shortest dependency paths (SDP) and a subtree of the lowest common ancestor (LCA), are generally employed. Fig. 1 shows a real case where decisive information is ignored when the model is restricted to only considering the SDP or the LCA subtree.

To resolve the aforementioned issues, we propose a novel Attention-based Graph Convolution Networks (AGCN) for DDI extraction, which is inspired by previous work (Zhang, Qi, & Manning, 2018b). We adopt GCN (Kipf & Welling, 2017) to encode the syntactic dependency graphs, producing latent feature representations of nodes (e.g., words in our case). By stacking the convolution layer, the model can capture richer neighborhood information of the graph. The model differs from other tree-based models such as Tree-LSTM, in that the GCN can be effectively applied over a dependency graph in parallel and do not constrain the input structure which can be any linguistic feature represented as a graph. We also consider contextualized information as well as syntactic knowledge by incorporating the GCN with recurrent networks such as bi-directional long short-term memory (bi-LSTM) networks.

In addition, we utilize full dependency trees as inputs to avoid the loss of crucial information. Our objective is to optimally use the relevant information while ignoring irrelevant information, in contrast to rule-based pruning. Thus, we can alleviate the loss of important clues in the full trees. Specifically, we developed an attention-based pruning strategy that assigns attention weights to each edge via a self-attention mechanism (Vaswani et al., 2017) to represent the strength of relatedness between nodes. This allows the model to learn whether to include or exclude information from the full tree. Consequently, the AGCN can exploit the context and structure of the input sentence as efficiently as possible.

We trained and evaluated our AGCN model on the dominant DDI extraction dataset of the DDIExtraction 2013 shared task (SemEval-2013 Task 9) (Segura-Bedmar, Martínez, & Herrero-Zazo, 2013), which consists of a corpus from the DrugBank database and MEDLINE abstracts. The proposed model achieved a micro *F*-score of 76.86%, outperforming the state-of-the-art model for DDI extraction by 1.38%. We also evaluated our model components with regard to their effectiveness and contribution to the performance improvement.

The main contributions of this paper can be summarized as follows:

- We propose a neural framework AGCN, which is the first graph convolution-based architecture for DDI extraction, to the best of our knowledge. We leverage both contextual and syntactic information regarding the sentence, which have complementary characteristics, by incorporating the GCN with recurrent networks.
- We introduce a novel attention-based pruning strategy to efficiently utilize syntactic information while ignoring irrelevant information, in contrast to previous rule-based pruning methods.
- The proposed framework achieved state-of-the-art results for DDI extraction with a micro *F*-score of 76.86% for the DDIExtraction 2013 dataset, outperforming the existing approaches. Furthermore, we conducted experiments using several types of RNN modules, pruning strategies, and linguistic features for comparison.

The remainder of this paper is organized as follows. In Section 2, we review existing approaches. In Section 3, we describe our AGCN model in detail. Sections 4 and 5 present the experimental setup and results, respectively. Finally, we present our conclusions in Section 6.

2. Related works

The existing approaches for DDI extraction can primarily be divided into three main categories: feature-based, kernel-based, and neural network-based. Feature-based approaches concentrate on finding distinctive features representing characteristics of the data. Various linguistic features are extracted and fed to classifiers trained on these features. For instance, Björne, Kaewphan, and Salakoski (2013) exploited shortest path features and domain knowledge features, and (Chowdhury & Lavelli, 2013a) utilized heterogeneous features consisting of semantic, lexical, syntactic, and negation features derived from parse trees. Similarly, Kim et al. (2015) combined word, parse tree and dependency graph features, and (Raihani & Laachfoubi, 2016) integrated lexical, phrase and phrase auxiliary features to extract DDI from the biomedical literature. In these methods, the major challenge is to identify the informative and suitable features, and the feature extraction process is also time-consuming and dependent on domain experts.

Kernel-based approaches depend largely on the designed kernels, which summarize the data instances and calculate their similarity. In general, these methods are known to take advantage of syntactic information, including dependency graphs and parse trees, compared with feature-based methods. Zhang, Lin, Yang, Wang, and Li (2012) proposed a single kernel-based method to effectively utilize syntactic information with a dependency graph.

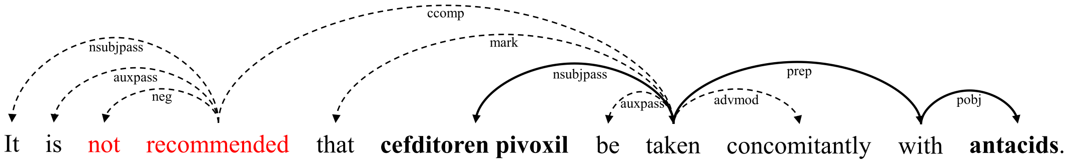

ccomp
nsubjpass
mark
auxpass
nsubjpass
prep
neg
auxpass
advmod
pobj
It is not recommended that cefditoren pivoxil be taken concomitantly with antacids.

Fig. 1. Example of a dependency tree for a sample sentence in DDI extraction task. The edges represent neighbor tokens $K = 1$ away. The shortest dependency path between drug entities is highlighted in bold. The root node of the LCA subtree of the entities is "*taken*". Note that a negative "*not*" and "*recommended*", which are crucial clues for DDI extraction, are excluded from both the shortest dependency path and the LCA subtree.

(Chowdhury & Lavelli, 2013b) introduced a hybrid kernel method that employed three different kernels and used contextual and shallow linguistic features, and they ranked at the top of the DDI extraction 2013 challenge. Thomas, Neves, Rocktäschel, and Leser (2013) achieved second place in this competition; they developed an ensemble-based model applying multiple kernel methods. However, a potential disadvantage of kernel methods is that all data structures are comprehensively represented by the kernel; thus, designing elaborate kernel functions is essential.

In contrast, deep neural networks have emerged for automatic representation learning methods and have exhibited remarkable performance in a wide range of fields, such as image processing, natural language processing, and information retrieval. In DDI extraction, neural network-based models have become the dominant method. Features for DDI prediction can be learned and extracted automatically using neural networks, without laborious feature engineering. There are two ways to extract relations between entities using neural networks, according to the input structure: sequence-based and dependency-based.

Sequence-based models primarily exploit diverse neural architectures, including CNN and RNN. Liu, Tang, Chen, & Wang, (2016a) introduced CNN model to predict DDI and combined word embeddings with position embeddings. The position embeddings were used to encode the relative distances between two entities and were widely employed in subsequent studies. Quan, Hua, Sun, and Bai (2016) developed a multichannel CNN model, where different versions of word embeddings were integrated to better represent the input sequences. RNN-based methods have been successfully employed to extract DDI as well. LSTM and gated recurrent units (GRU)—a kind of RNN-based architecture—have been used by Zhou et al. (2018), Yi et al. (2017) and Sahu and Anand (2018), and they achieved better performance than CNN-based methods. Furthermore, Sun et al. (2019) proposed a recurrent hybrid CNN to exploit both semantic and sentence-level representations. This approach yielded an *F*-score of 75.48%, which is the best performance achieved thus far for the DDIExtraction 2013 corpus.

Dependency-based approaches focus on incorporating structural information of given sentences into the neural architectures. The DDI extraction corpus is composed of several long and complex sentences. The longest sentence contains more than 150 words, presenting a significant challenge. Therefore, structural knowledge, e.g., dependency trees, is useful for DDI prediction. Zhao et al. (2016) showed that dependency paths are effective in neural models for identifying DDI, and (Liu, Chen, Chen, & Tang, 2016b) introduced dependency-based CNN. In particular, Zhang et al., (2018a) exploited both sentence sequences and dependency paths via hierarchical RNN. This approach is along the same lines of ours, in that it integrates semantic and syntactic information. However, all previous studies on DDI extraction, including those mentioned above, have focused only on the SDP.

A variety of dependency pruning techniques have been employed to reduce computational costs and improve the performance of natural language processing. One common pruning strategy involves the SDP between two entities in the full path described above (Xu et al., 2015), and another common approach is to exploit the subtree below the LCA of the entities (Miwa & Bansal, 2016). Zhang, Qi, & Manning, (2018b) proposed a path-centric pruning method that includes tokens up to a distance *K* away from the dependency path in the LCA subtree. However, these rule-based pruning techniques risk excluding valuable information from the original tree for relation extraction. In contrast to previously reported methods in which rule-based preprocessing is adopted to eliminate edges, our approach can automatically learn and assign different weights to each edge according to its relative importance.

Graph convolutional networks (GCN) (Kipf & Welling, 2017) have been successfully employed to generalize neural networks that operate on arbitrarily structured graphs, including knowledge graphs, social networks, and dependency graphs. We can obtain insight regarding the relationships between the entities (i.e., nodes) by representing data as a graph structure. In early works (Gori, Monfardini, & Scarselli, 2005; Scarselli, Gori, Tsoi, Hagenbuchner, & Monfardini, 2009), researchers proposed a type of graph neural networks (GNN) to address more general graphs, such as cyclic, undirected, and directed graphs. To improve the computational efficiency, Duvenaud et al. (2015) introduced CNN that operate directly on graphs of arbitrary size and shape for graph classification. Kipf and Welling (2017) simplified this approach by restricting the filters to operate on a first-order neighborhood around each node, producing representations that encode both the local graph structure and the features of nodes.

In detail, a number of recent studies for the general GNN architectures that also exploit RNN have been conducted, similar to our approaches incorporating GCN with RNN. Li, Zemel, Brockschmidt, and Tarlow (2016) addressed the problem of predicting the sequence of outputs such as paths on a graph from a single input graph. In this approach, the model extended the GNN model through the use of GRU to generate multiple sequences. Seo, Defferrard, Vandergheynst, and Bresson (2018) introduced graph convolutional recurrent networks that also predict the sequences of graph-based data. They merged GCN and RNN to simultaneously exploit both the spatial structures of graphs and dynamic information about the data. Manessi, Rozza, and Manzo (2020) presented dynamic graph convolutional networks to deal with dynamic graph-structured data that may change over time in the real world. This model combined LSTM and GCN to capture long short-term dependencies together with the graph structure. These approaches exploited GNN in conjunction with RNN architecture similarly to ours, but they differ in their purpose and combination method. All of them employed RNN to extract temporal information from graphs, which can be composed of multiple sequences or appear dynamic, and our model handles contextual information from sentences. Additionally, we focus on feature representation for static graph-structured inputs by leveraging an attention-based pruning technique and produce a single output such as DDI information. Consequently, the proposed approach is capable of effectively generating the representation suitable for the task.

The attention mechanism has recently achieved remarkable success in natural language processing tasks such as machine translation (Bahdanau, Cho, & Bengio, 2015; Luong, Pham, & Manning, 2015) and question answering (Seo, Kembhavi, Farhadi, & Hajishirzi, 2017). The basic concept of the attention mechanism is to pay more attention to context that is the most relevant at a specific point when models need to make decisions. For DDI extraction, RNN model with multiple attention layers for word- and sentence-level attention has been proposed (Yi et al., 2017). Zhou et al. (2018) demonstrated that a position-aware attention mechanism combining position embeddings with the hidden states of bi-LSTM is effective for predicting the interaction between drugs. Our method differs from previously reported ones in that our attention mechanism operates on GCN architecture, leveraging the self-attention mechanism relating different nodes of a single dependency graph to automatically exploit the important tokens among all the tokens.

3. Proposed method

Consider the problem of each sentence $X = [x_1, \cdots, x_n]$, where $x_i \in \mathbb{R}^d$ represents the *d*-dimensional *i*th embedded token. Drug entities, which are denoted as DRUG-A and DRUG-B, are identified and correspond to the words x_a and x_b $(a, b \in [1, n], a \neq b)$ in the sentence. Given X, x_a, and x_b, the proposed model (AGCN) predicts a relation $r \in \mathscr{R}$ (a predefined relation set) that holds between the drug entities, or "no-relation" otherwise. Fig. 2 illustrates an overview of the proposed framework. First, each word is represented as an embedded token that contains a word embedding, dependency embedding, part of speech (POS) embedding, and distance embedding. These representations are sequentially fed into the bi-LSTM and GCN to capture both contextual and syntactic features. Then, our framework prunes irrelevant information while making optimal use of the meaningful information via the proposed attention-based pruning method. In the pooling layer, a sentence representation and two drug representations are obtained, and these representations are concatenated. Finally, the classifier extracts the types of interactions between drug entities.

In this section, we first describe basic graph convolutional networks (GCN) for DDI relation extraction, and then describe our Attention-based Graph Convolution Networks (AGCN) in detail.

3.1. Graph convolutional networks

Graph convolutional networks (GCN) (Kipf & Welling, 2017) is a type of convolutional neural networks that operate directly on graphs. We adopt the GCN to model the dependency tree converted into the graph structure. The GCN model encodes information about the neighborhood of each node as a feature vector, sharing filter parameters over all locations in the graph. The convolution operation in the GCN is similar to that in the CNN in that the model shares parameters in kernels. In each layer, each node gathers and summarizes information from its all immediate neighbors; thus, information is conveyed along the edges of the graph. Depending on the number of convolution layers, the model can encode rich neighborhood information of the graph. For instance, K layers GCN can capture information about neighbors a maximum of K hops away.

Formally, consider an undirected graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ represents a set of n nodes, and $\mathscr{E}$ represents a set of edges. The objective is to learn a function of features on the graph that takes as input: an adjacent matrix A and an input feature $h_i^{(l)}$. The adjacent matrix A represents the graph structure with $n \times n$ nodes, where $A_{ij} = 1$ if there is an edge going from node i to node j. The convolution operation for node i at the l^{th} layer takes $h_i^{(l-1)}$ as an input. The node representation $h_i^{(l)}$ about its neighbors is computed as:

$$h_i^{(l)} = \sigma\left(\sum_{j=1}^{n} A_{ij} W^{(l)} h_j^{(l-1)} / d_i + b^{(l)}\right), \tag{1}$$

where $W^{(l)}$ and $b^{(l)}$ represent a kernel weight matrix and a bias term, respectively, and σ represents a nonlinear activation function (e.g., ReLU). The initial state h_i^0 takes x_i as an input. In our AGCN, h_i^0 takes contextualized features obtained from the bi-LSTM model. We add a self-loop to each node in order to incorporate the node itself by adding the identity matrix I to the adjacent matrix. Additionally, we assume that the dependency graph is undirected, which means $A_{ij} = 1$ and $A_{ji} = 1$. A normalization term $d_i = \sum_{j=1}^{n} A_{ij}$ represents the degree of the token. Note that A is reconstructed to the weight of each edge for attention-based pruning in our method.

3.2. Context-sensitive representations

The introduced GCN model is capable of effectively capturing dependencies between nodes for various tasks (Zhang, Qi, & Manning, 2018b; Bastings et al., 2017). However, the network considers only syntactic features, which leads to a lack of lexical and contextual features from the original sentence. The same word may have diverse meanings in different contexts, making it difficult for the word vector to accurately express contextual meaning. Because the pre-trained word embeddings that are widely used only allow a single context-independent representation for each word, polysemy is challenging. We hypothesize that context-sensitive vectors will facilitate the extraction of the semantic relations from the sentences. Therefore, we expect the contextual and syntactic information to play complementary roles, enriching their states.

To resolve these issues, we employ bi-directional LSTM (bi-LSTM). For word token x_i, the two LSTM layers capture contextual information along the sentence, both forward and backward. The networks read the previous and future context of the current time step. Thus, the bi-LSTM model identifies the context-dependent meaning of the word better than does the one-way

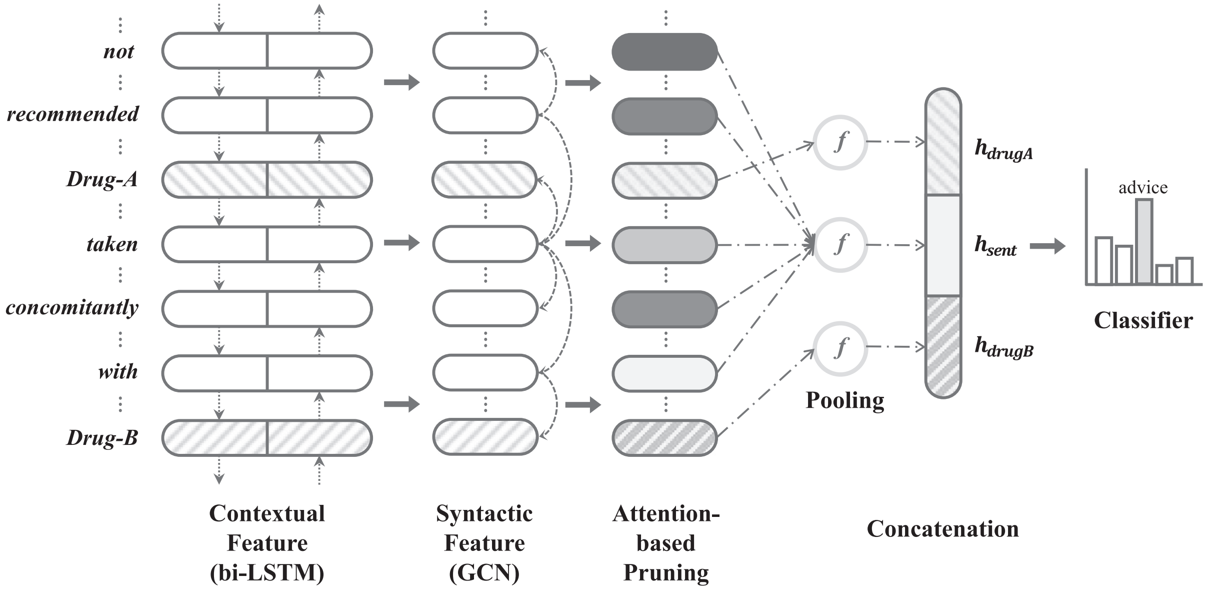


Fig. 2. Overview of the proposed Attention-based Graph Convolutional Networks architecture.

network. In our model, the input word vectors are first fed into the bi-LSTM, and the outputs $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ of the bi-LSTM at the last time step i are then concatenated into h_i, as follows:

$$\overrightarrow{h_i} = LSTM\left(x_i, \overrightarrow{h_{i-1}}\right) \tag{2}$$

$$\overleftarrow{h_i} = LSTM\left(x_i, \overleftarrow{h_{i-1}}\right) \tag{3}$$

$$h_i = \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right]. \tag{4}$$

The output h_i, i.e., the context-sensitive features, are employed as the initial state h_i^0 in the AGCN.

3.3. Attention-based pruning

Dependency trees convey syntactic information about the sentence, which is valuable for relation extraction (Fundel, Küffner, & Zimmer, 2007). However, most existing dependency-based networks do not take the full dependency trees. The dependency trees are directly pruned by eliminating irrelevant information from the original tree, because of the equivalent weights of the binary adjacent matrix. As discussed in Section 2, such pruning strategies are generally predefined in rule-based preprocessing and can cause a loss of crucial information and performance degradation. Motivated by these observations, we propose *attention-based pruning*, where all of the edges in the full dependency graph are assigned to the weights.

Therefore, we reconstruct the adjacent matrix A into a soft adjacent matrix $\widehat{A}$ with a self-attention mechanism (Vaswani et al., 2017). The self-attention mechanism captures the relation between different positions of a single sequence, which has led to successful results in a variety of tasks including reading comprehension and summarization (Lin, Sun, Ma, & Su, 2018; Yu et al., 2018). We thus employ the self-attention mechanism to determine the relatedness between nodes, even if they are connected indirectly. The existing pruning methods are regarded to assign zero weights to each edge if the two nodes are not connected directly. In contrast, we assign the weights to all the edges, which is viewed as a soft-attention approach.

We compute the attention function on query and key pairs of d-dimensional vectors representing tokens. The output weights are calculated using a compatibility function of the query with the corresponding key, and then assigned to the values. To capture a different context from multiple aspects, we found it more beneficial to employ multi-head attention, similar to Vaswani et al. (2017). Specifically, H independent attention mechanisms are executed, and their output features are concatenated, which allows the model to jointly attend to information from different representation subspaces. We compute the soft adjacent matrix as:

$$\widehat{A}_{\mathrm{h}} = softmax\left(\frac{QW_h{}^Q \times \left(KW_h{}^K\right)^T}{\sqrt{d}}\right), \tag{5}$$

where Q and $K$are equal to $h^{(l-1)}$, i.e., the representation at the previous convolutional layer. The projection parameters are $W_h{}^Q \in \mathbb{R}^{d\times d}$ and $W_h{}^K \in \mathbb{R}^{d\times d}$, where h refers to the hth head in H attention layers. Accordingly, $\widehat{A}_h$ represents the hth soft adjacent matrix and we can obtain H soft adjacent matrices from the binary adjacent matrix. In practice, the binary adjacent matrix is firstly used for the initial node representation h_i^0, and we apply the soft matrices from later convolutional layers.

Each soft matrix is employed for a convolution operation. Hence, we have H node representations $\left\{h_{i_k}^{(l)}\right\}_{k=1}^{H}$, which are concatenated and linearly transformed. This leads to the integration of various meanings of multi-head attention into one vector, which has the same dimension as the input feature through the projection layer. In summary, we calculate each layer as follows:

$$h_{i_k}^{(l)} = \sigma\left(\sum_{j=1}^{n} \widehat{A}_{ij_k} W_k^{(l)} h_{j_k}^{(l-1)} / d_i + b_k^{(l)}\right) \tag{6}$$

$$h_i^{(l)} = Linear\left(\left[h_{i_1}^{(l)}; \cdots; h_{i_H}^{(l)}\right]\right), \tag{7}$$

where $k = 1, \cdots, H$, and the weight matrix $W_k^{(l)}$ and bias term $b_k^{(l)}$ depend on k. *Linear* refers to a linear projection layer. The procedure thus far for attention-based pruning is summarized in Fig. 3. Because we have a dimensional representation of the original input feature, the corresponding representation is fed into the next layer as the input feature. After the last layer, we apply a one simple convolutional layer to induce the final representation of each node. This layer leaves the adjacent matrix out because the input features already contain the structure information from the previous process. We calculate the final representation as:

$$h_i^{(\overline{L})} = \sigma\left(\sum_{j=1}^{n} W^{(L)} h_j^{(L)} + b^{(L)}\right). \tag{8}$$

3.4. Extracting DDI with AGCN

Given a sentence $\mathrm{X} = [x_1, \cdots, x_n]$ in which x_a = "DRUG-A" and x_b= "DRUG-B", each word x_i is represented as four linguistic features: the word itself, part of speech (POS) tag, dependency and distance. Each input word is mapped to a pre-trained word embedding. We exploit the POS and dependency feature of the word to extend its representation ability. As the dependency relation between a head word and a child can lead to a difference in meaning, the dependency feature is equipped to capture such grammatical relations (Xu et al., 2015). In particular, the POS feature is informative for distinguishing different semantic meanings in sentences (Zhang et al., 2018a; Zhao, Yang, Luo, Lin, & Wang, 2016). The distance feature proposed by Zeng, Liu, Lai, Zhou, and Zhao (2014) represents the relative distance between x_i and the target drugs x_a and x_b, respectively. It is helpful to specify which input words are the two target nouns in the sentence. Thus, two distance measures are defined for each word x_i. Indeed, we compare the effects of these features through an ablation study in Section 5.4. Let E^{word}, E^{pos}, E^{dep}, E^{dis_1}, and E^{dis_2} denote the word embedding matrix, POS embedding matrix, dependency embedding matrix and two distance embedding matrices, respectively. We randomly initialize and fine-tune E^{pos}, E^{dep}, E^{dis_1}, and E^{dis_2} during training. The final word representation is given by $x_i = \left[e^{word}; e^{pos}; e^{dep}; e^{dis_1}; e^{dis_2}\right]$, where ';' represents the concatenation operator.

After applying our AGCN model over the input dependency graph, we obtain hidden representations of each token, which is affected by its neighbor nodes K hops away. Our objective is to extract the interaction between the drug pair using these representations. We produce a sentence representation and two drug representations, respectively, as follows:

$$h_{sent} = f\left(mask_{sent}\left(h^{(\overline{L})}\right)\right) = f\left(\mathrm{AGCN}\left(h^{(0)}\right)\right), \tag{9}$$

where $h^{(l)}$ denotes the collective representations in the lth layer of the AGCN, and $f :\in \mathbb{R}^{d\times n} \rightarrow \mathbb{R}^d$ is a max pooling function that maps n output vectors to one sentence vector. The function $mask_{sent}$ selects

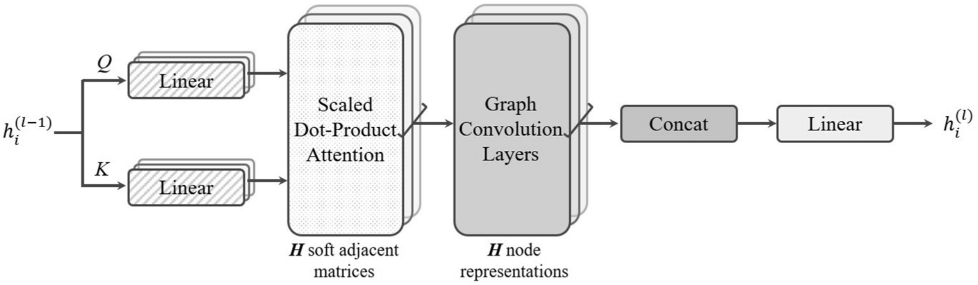


Fig. 3. Diagram of the attention-based pruning strategy.

only sentence representations except for the entity tokens. We also obtain two drug representations h_{drugA} and h_{drugB} in a similar way.

Finally, we obtain the final representation for DDI prediction by concatenating these representations and feeding them to a fully connected layer with the activation function, following (Lee, He, Lewis, & Zettlemoyer, 2018; Santoro et al., 2017):

$$h_{final} = FClayer\left(\left[h_{sent}; h_{drugA}; h_{drugB}\right]\right). \tag{10}$$

Here, h_{final} is taken as the input into a linear layer, followed by a softmax classifier to generate a probability distribution over the DDI relations.

4. Experiments

4.1. Dataset

In our experiments, we evaluated the proposed model for the DDIExtraction 2013 dataset (Segura-Bedmar et al., 2013), which is the most widely known dataset for evaluating the performance of DDI extraction. The DDIExtraction 2013 corpus is manually annotated and is composed of the DrugBank and MEDLINE abstracts. The main purpose of this task is to detect the drug-drug interactions and classify each DDI into one of five distinguishable DDI types: Advice, Mechanism, Effect, Int, and Negative. We briefly describe each type with an example.

- Advice: Advice is assigned when the text provides a recommendation or advice regarding the concomitant use of two drugs; e.g., "*Concomitant use of bromocriptine mesylate with other ergot alkaloids is not recommended.*"
- Effect: Effect is assigned when the text mentions a pharmacodynamic mechanism such as a clinical finding, increased toxicity, or therapeutic failure; e.g., "*Therefore, a slower onset can be anticipated if STADOL NS is administered concomitantly with, or immediately following, a nasal vasoconstrictor.*"
- Mechanism: Mechanism is assigned when the text mentions a pharmacokinetic mechanism such as changes in the levels or concentrations of the drugs; e.g., "*Probenecid competes with meropenem for active tubular secretion and thus inhibits the renal excretion of meropenem.*"
- Int: Int is assigned when the text mentions an interaction but does not provide any additional information about the interaction; e.g., "*A two-way interaction between the hydantoin antiepileptic, phenytoin, and the coumarin anticoagulants has been suggested.*"
- Negative: Negative is assigned when the text mentions no interaction between the two drugs; e.g., "*The safety and efficacy of PROLEUKIN in combination with any antineoplastic agents have not been established.*"

The dataset is split into two parts: training data for development, and testing data for evaluation. We first performed standard preprocessing steps, such as tokenizing and normalizing on both the training and test data. These steps help to reduce the size of the vocabulary and improve the performance. For tokenizing, we employed the GENIA tagger (Tsuruoka et al., 2005), which was specifically tuned for the biomedical corpus such as MEDLINE abstracts. We did not eliminate little distinctive words such as prepositions and conjunctions because we needed to obtain the dependency graph from complete sentences. We also changed each digit that was not a substring of a drug entity to a special tag '#'. In particular, we anonymized target drugs to generalize our approach, following a previously reported method (Zhao et al., 2016). In their study, the drug names did not play a major role in extracting DDI. Thus, for sentences with more than two drugs, we replaced the two target drug entities with the symbols "DRUG-A" and "DRUG-B", while all other drug entities were represented as "DRUG-N". However, such a simple preprocessing strategy may produce an imbalanced dataset, degrading the performance. For instance, the sentence "If replacing $drug_1$ by $drug_2$ therapy, the introduction of $drug_3$ should be delayed for several days after $drug_4$ administration has stopped." contains one positive instance ($drug_3$, $drug_4$) and five negative instances. The number of negative instances is significantly larger than the number of positive instances.

To reduce the number of negative instances, rule-based negative instance filtering is generally employed. We used data to which negative instance filtering was applied in a previous study (Zhao et al., 2016), for comparison. We briefly describe the filtering rules applied to the data, as follows. The first rule is to remove the instances with two target drugs referring to the same drug. The second rule is to filter instances with two target drugs connected with a coordinate structure (e.g., "and", "or" and a comma). The detailed statistics of the preprocessed DDI extraction dataset are listed in Table 1.

4.2. Experimental setting

In our experiments, we implemented our AGCN model with the PyTorch library. We used the biomedical word embeddings (Pyysalo, Ginter, Moen, Salakoski, & Ananiadou, 2013), which were pre-trained using unlabeled biomedical texts from PubMed and PubMed Central (PMC). We used the Stanford parser to obtain the dependency tree, dependency label, and POS tag of each word in the candidate sentence. The dimensions of the word, depen-

Table 1
Statistics of the datasets used in the experiments.

	Relation type	Training	Test
Positive	Advice	814	221
	Effect	1592	357
	Mechanism	1260	301
	Int	188	92
Negative		8987	2049
Total		12,841	3020

dency, POS, and distance embeddings are set at 200, 20, 20, and 15, respectively. The dimensions of the bi-LSTM and attention were 300. All models were trained with a batch size of 32 instances and the neural networks were optimized with stochastic gradient descent. To alleviate the overfitting problem, the L2 regularization weight was set to 0.003 in the output layer. Dropout, in which units and their connections are randomly dropped from the networks during training, was applied to 0.5 in the bi-LSTM, convolution, and attention layers. Additionally, we experimentally employed a two-layer architecture with which we achieved the optimal performance.

4.3. Evaluation metric

To evaluate the performance of the proposed model, we use the micro-precision (*micro-P*), micro-recall (*micro-R*) and micro-*F* score (*micro-F*), which have been employed for existing models. The micro-averaged metrics aggregate the contributions of all classes to calculate the average metric, which is appropriate for imbalanced data. The metrics are defined as follows:

$$micro - P = \frac{\overline{TP}}{\overline{TP} + \overline{FP}} \tag{11}$$

$$micro - R = \frac{\overline{TP}}{\overline{TP} + \overline{FN}} \tag{12}$$

$$micro - F = \frac{2 \times micro - P \times micro - R}{micro - P + micro - R} \tag{13}$$

where $\overline{TP}$, $\overline{FP}$, and $\overline{FN}$ denote the average scores calculated across different classes of true positives, false positives, and false negatives, respectively. Suppose that a dataset has *A* positive relation samples, and a relation extraction system can extract *B* positive relation samples. TP represents the case in which only some instances of predicted *B* instances are correctly predicted as positive. FP represents the case in which the system may incorrectly extract some relation instances as positive. Similarly, FN represents the case in which some relation instances in *A* are not extracted by the system. The precision is the ratio of the number of correctly predicted DDI to the total number of identified DDI. The recall is the ratio of the number of correctly predicted DDI to the total number of DDI that exist. The *F*-score is a harmonic mean of the precision and recall which indicates the overall performance of the DDI extraction system. Hence, *micro-F* is the harmonic mean of *micro-P* and *micro-R*, where *micro-P* and *micro-R* represent the precision and the recall, respectively, averaged over all samples and label pairs.

5. Results and discussions

We evaluated the proposed model in comparison with four conventional techniques and eight deep learning models for relation extraction. We first investigated the detailed performance of each DDI type for all methods and overall performance score. Additionally, we conducted an extensive experiment to examine the effectiveness of each component of the proposed method, including the contextualized representations and the pruning strategies. Furthermore, we investigated various embedding features to analyze the effects of each exploited feature.

5.1. Overall comparison

We selected four feature-based methods as traditional approaches that are not neural methods. Our feature-based baselines are widely used machine learning approaches for DDI relation extraction. These methods commonly utilize carefully handcrafted features and conventional classifiers such as SVM. The characteristics of the approaches are as follows:

- UTurku (Björne et al., 2013) uses features from dependency parsing and domain resources such as DrugBank.
- FBK-irst (Chowdhury & Lavelli, 2013b) combines linear features, path-enclosed tree kernels, and shallow linguistic features.
- Kim et al. (2015) use lexical, semantic, contextual, and tree-structured features all together.
- Raihani and Laachfoubi (2016) use diverse features and rules that are elaborately designed for the SVM classifier of each subtype.

Additionally, we compared our method with other neural network-based approaches, including state-of-the-art models for DDI relation extraction. Neural methods automatically learn the high-level feature representations based on the different architectures:

- CNN (Liu, Tang, Chen, & Wang, 2016a) is a basic convolutional neural network with the word and position embeddings.
- SCNN (Zhao et al., 2016) is a syntax convolutional neural network with a word embedding that contains the syntactic information based on the shortest dependency path.
- MCCNN (Quan et al., 2016) is a multichannel convolutional neural network, where different versions of word embeddings are integrated to better represent each word.
- Joint-LSTM (Sahu & Anand, 2018) is an ensemble model comprising classical LSTM and attention-based LSTM.
- GRU (Yi et al., 2017) is a recurrent neural network model with word-level and sentence-level attention layers.
- Hierarchical RNN (Zhang et al., 2018a) is a hierarchical LSTM network that exploits the shortest dependency path and the sentence sequence.
- PM-BLSTM (Zhou et al., 2018) is an attention-based LSTM network with a position-aware attention layer.
- RHCNN (Sun et al., 2019) is a joint model of the recurrent network and hybrid convolutional network with a focal loss.

Table 2 presents the experimental results for the baselines and our method. We cite experiment results from the original paper for each model. We did not report the scores if the original paper did not contain detailed or overall results. The entire methods were evaluated with the same training and test datasets used for our proposed model. As indicated by the table, we computed the *F*-score for four DDI types, as well as the overall precision, recall, and *F*-score. The neural network-based models, including the proposed method, exhibited significantly better performance than the traditional approaches with a large margin. Because the traditional methods depend strongly on the input features, feature engineering to retrieve information from the input sentence is crucial. However, manually generated features are not sufficient compared with the neural methods for determining the interaction between drugs, because of the structural complexity of natural language.

The AGCN achieved scores of 78.17%, 75.59%, and 76.86% with respect to the precision, recall, and *F*-score, respectively, and the best scores are emphasized in bold. Our model resulted in the highest overall performance scores among all the neural network-based models, including the previous state-of-the-art model, except for the *Int* type. In the case of the *Int* type, there is a severe data imbalance in the DDIExtraction 2013 dataset, as shown in Table 1, which led to poor performance for all the models. Nevertheless, our model consistently exhibited higher results than the existing methods, with regard to the overall scores and the detailed scores, except for the *Int* type. To address the problem of data

Table 2
Performance comparison with other state-of-the-art methods.

	Methods	*F*-score for each DDI type (%)				Overall (%)		
		Advice	Effect	Mechanism	Int	Precision	Recall	*F*-score
Traditional methods	UTurku	63.00	60.00	58.20	50.70	73.20	49.90	59.40
	FBK-irst	69.20	62.80	67.90	54.70	64.60	65.60	65.10
	Kim et al.	72.50	66.20	69.30	48.30	–	–	67.00
	Raihani et al.	77.40	69.60	73.60	52.40	73.70	68.70	71.10
Neural network-based methods	CNN	77.72	69.32	70.23	46.37	75.70	64.66	69.75
	SCNN	–	–	–	–	72.50	65.10	68.60
	MCCNN	78.00	68.20	72.20	51.00	75.99	65.25	70.21
	Joint-LSTM	79.41	67.57	76.32	43.07	73.41	69.66	71.48
	GRU	–	–	–	–	73.67	70.79	72.20
	Hierarchical RNN	80.30	71.80	74.00	54.30	74.10	71.80	72.90
	PM-BLSTM	81.60	71.28	74.42	48.57	75.80	70.38	72.99
	RHCNN	80.54	73.49	78.25	58.90	77.30	73.75	75.48
Our method		**86.22**	**74.18**	**78.74**	52.55	**78.17**	**75.59**	**76.86**

imbalance, we attempted to experiment with the *Adjusted F-measure* reported by Maratea, Petrosino, and Manzo (2014), which is a measure for the classification performance in the case of data imbalance that provides more weight to patterns correctly classified in the minority class. Using the *Adjusted F-measure*, we achieved scores of 87.17%, 86.38%, 85.15%, and 83.06% with respect to the AGCN, RHCNN, PM-BLSTM, and CNN, respectively. As the result of the experiment, we found that the experimental results based on the *Adjusted F-measure* show similar aspects to those based on the *F*-score, and the AGCN also demonstrated the best performance.

These results indicate that the graph structure of sentences can be helpful for inferencing the relations in sentences. Because the AGCN effectively encodes the dependency structures of sentences through GCN using attention-based pruning, our model explicitly detects relations between two drugs for a given sentence. The baselines primarily employ CNN or RNN variants to encode the sentence information implicitly, which limits the path between entities. On top of that, our model learns to optimize the path, i.e., the adjacent matrix, via the trainable attention matrix. To minimize the loss of information in the graph, the soft adjacent matrix plays an important role in the AGCN.

Additionally, we fuse contextual representations from the bi-LSTM, similar to RHCNN whereas the other methods use only the word embeddings. As the semantic meaning of a word depends on the context of the sentence as well as the word itself, the model takes informative knowledge from the contextualized representation. The detailed contribution of each component is discussed in the following section.

5.2. Effect of contextualized representations

We conducted an additional experiment to demonstrate the effectiveness of the contextualized information and the different RNN models. Table 3 presents the experimental results. To investigate the impact of the contextualized representations, we first utilized a basic GCN that removes the RNN layers for context-sensitive features in the AGCN. The attention-based pruning strategy was applied in all the experiments reported in Table 3. Without the RNN layers, the *F*-score decreased by 3.00%, as shown in the first row of the table. As previously discussed, the meaning of words depends on their context of use. If we use only word embeddings, the semantics of words can be incorrectly represented. Therefore, by combining contextual information passed through RNN layers, we can obtain richer representations that reflect the contextual information of words.

Table 3
Experimental results for different RNN models with respect to contextualized representations.

Strategy	Precision	Recall	*F*-score
Basic GCN	74.84	72.91	73.86
+ GRU	75.66	73.33	74.48
+ LSTM	77.50	74.15	75.79
+ bi-LSTM	78.17	75.59	76.86

Next, we compared different RNN architectures, including GRU and LSTM. The GRU model employs the gating mechanism in the RNN to allow each recurrent unit to capture dependencies over different time scales. The LSTM model exploits memory cells as well as the gating mechanism to alleviate the long-term dependency problem. The performances of GRU, LSTM, and bi-LSTM for our model are listed in Table 3. When we used GRU and LSTM, we achieved the *F*-scores of 74.48% and 75.79%, respectively. When we applied the bi-directional LSTM layer, we achieved the highest *F*-score, 76.86%. This result suggests that simultaneously obtaining information from the backward and forward states is more useful than a one-way network for representing the context-dependent meaning of words.

5.3. Effect of attention-based pruning

To evaluate the effectiveness of the attention-based pruning strategy, we compared it with two other strategies: a full tree without pruning and the LCA-based pruning method proposed in (Zhang, Qi, & Manning, 2018b). In the case of the LCA subtree, k represents the dependency tree that includes tokens up to distance k away in the LCA subtree. We conducted the experiment on our AGCN model with different pruning strategies for fair comparison. As shown in Table 4, we observed that the proposed attention-based pruning approach exhibited better performance than the other strategies. Our approach obtained 3.67% better performance than the best performing model with a full tree. This verifies our hypothesis that incorporating relevant information is crucial for DDI extraction. Our strategy can effectively identify and extract more key words from the candidate sentences, which is vital for

Table 4
Experimental results for different pruning strategies.

Pruning strategy	Precision	Recall	*F*-score
Full Tree	73.99	72.39	73.19
LCA (k = 0)	70.62	69.82	70.22
LCA (k = 1)	72.81	71.16	71.97
LCA (k = 2)	73.79	72.20	72.98
Attention-based	78.17	75.59	76.86

dealing with long and complicated sentences to predict interaction pairs from biomedical text.

The results obtained using the LCA subtree demonstrate that utilizing more information while increasing the value of k is beneficial for DDI extraction. This indicates that relation extraction in the biomedical domain has slightly different aspects. In the general domain (Zhang, Qi, & Manning, 2018b), including extra information reduces the effectiveness, as indicated by the lowest performance when the entire dependency tree is included. In biomedical articles, the two target entities can be located close to each other because the drug entities are often enumerated in the sentence. Using the LCA or SDP between them might cause a significant loss of information. Therefore, our attention-based pruning strategy exhibited the best performance, taking advantage of the given sentence as efficiently as possible for DDI extraction. The results demonstrate the effectiveness of attention-based pruning for identifying DDI.

We conducted an additional experiment to determine the optimal number of heads for the multi-head attention that we employed to prune the dependency graph. We evaluated our AGCN model with different numbers of heads, i.e., $H \in \{0, 1, 2, 3, 4\}$ for the same environments. Table 5 presents the experimental results. As shown in the results, it is notable that all the models, even with different values of H, consistently outperformed the previous state-of-the-art model in Table 2. The performance of our model was the best with $H = 3$, with an F-score improvement of 0.89. When four or more heads were used, the F-score decreased slightly. Consequently, the experimental results indicate that with the multi-head attention mechanism, our model can capture different contexts from multiple aspects and improve the performance. Based on these observations, we experimentally set the number of heads (H) as 3 for our model.

5.4. Effect of different features on performance

In this section, to analyze the effects of the different types of linguistic features that we exploited, we evaluated models with four different combinations of features. Firstly, we confirmed that the pre-trained embeddings are slightly better than random initialization. Because it is difficult to learn truly representative features with insufficient data, we can previously obtain some semantic information through the pre-trained embeddings. We started with a baseline system without additional features except for the word embeddings, and then gradually added features. The contextualized representations and attention-based pruning were applied in all experiments, and the result are presented in Table 6. Our method achieved an F-score of 74.04% with only the word embeddings. When POS embeddings and dependency embeddings were integrated with the word embeddings, the performance was improved. After the distance features were additionally exploited, the performance was further improved with a total increase of 1.17% in the F-score (75.69% versus 76.86%). The results indicate the importance of distance features in DDI tasks for identifying target drug entities in long sentences and highlighting the key information within the sentences.

Table 5
Experimental results for the multi-head attention mechanism. The best scores are emphasized in bold.

# of heads	Precision	Recall	F-score
$H = 0$ (w/o attention)	73.99	72.39	73.19
$H = 1$	76.57	75.39	75.97
$H = 2$	77.18	75.58	76.38
$H = 3$	**78.17**	75.59	**76.86**
$H = 4$	76.30	**76.93**	76.61

Table 6
Experimental results for the different types of linguistic features.

Embedding features	Precision	Recall	F-score
Random initialization (w/o pre-trained embeddings)	72.41	74.87	73.61
Word	73.04	75.07	74.04
Word + POS	74.92	73.84	74.56
Word + POS + Dependency	76.85	74.56	75.69
Word + POS + Dependency + Distance	78.17	75.59	76.86

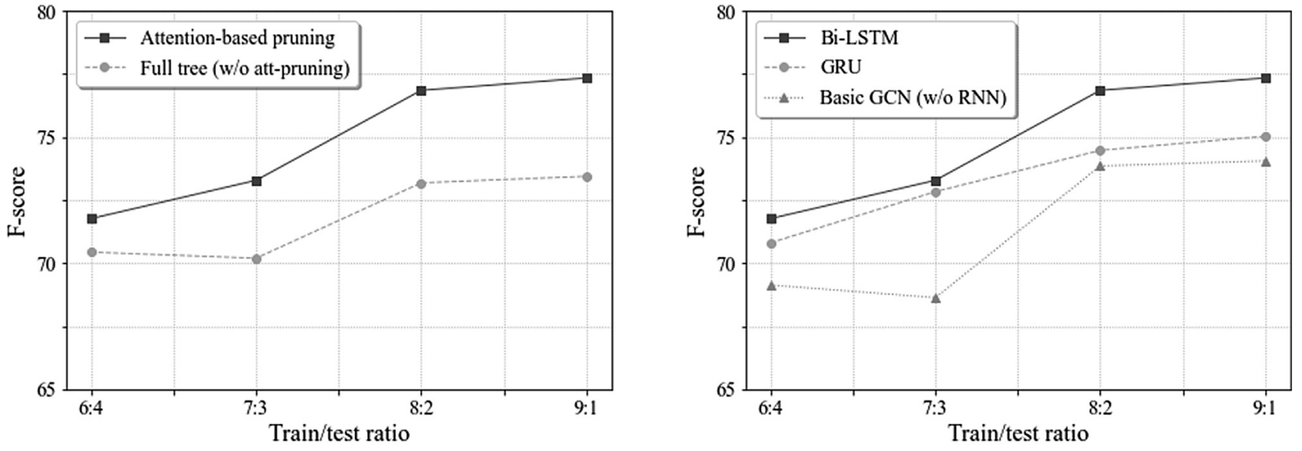
5.5. Effect of training data size on performance

In order to study the contribution of the size of the training dataset for boosting the performance, we compared four different training and test ratios. Because the DDIExtraction 2013 dataset after negative instance filtering is divided into training data and evaluation data at a ratio of approximately 8:2, we further experimented with ratios of 6:4, 7:3, and 9:1. We plotted the results in Fig. 4. As illustrated in the figure, the proposed model yields better performance as the amount of training data is increased, both with and without the attention-based pruning. Additionally, it is obvious that the contextual representations and attention-based pruning strategy are highly effective for the DDI task, as it outperforms the baseline by a large margin. This implies that the proposed model can learn how to extract and utilize informative relations in the dependency tree through the pruning strategy and can exploit contextual and structural knowledge by fusing the context-sensitive features. Apparently, the ability of the proposed method can be enhanced when there are more observations, hence the performance gap increases with higher training ratios.

6. Conclusion

In this paper, we propose an efficient and competitive method for DDI extraction, called the Attention-based Graph Convolutional Networks (AGCN). The proposed model primarily consists of two designs that contribute to its enhanced performance. First, the architecture utilizes both contextual and syntactic information together. Although dependency trees convey valuable syntactic information, existing approaches have predominantly focused on only the sequential or structural information of sentences. Therefore, we leverage syntactic dependency graphs representing the latent features of each node, as well as the context, by incorporating the GCN with recurrent networks. Second, our model applies a novel attention-based pruning strategy instead of rule-based pruning. The previous rule-based pruning strategies may exclude crucial clues for relation extraction. We accordingly utilize full dependency trees as inputs, while ignoring irrelevant information by employing a self-attention mechanism. In this way, the AGCN can exploit the context and structure of the input sentence as efficiently as possible.

We evaluated the performance of the proposed model for the DDIExtraction 2013 dataset and compared it with that of other competitive methods. The proposed method exhibited the best performance, outperforming state-of-the-art methods in the DDI extraction task. Overall, the experimental results indicate that the proposed method has the ability to minimize the loss of information, while designing informative representations that convey both context-sensitive and syntactic information. Therefore, we demonstrated that the proposed model is a competitive system for extracting DDI from biomedical literatures. Recent studies have reported other methods for producing contextualized representations based on language modeling. We believe that an attempt to obtain contextualized representations via language modeling approaches would make for an interesting study.



(a) Experimental results for pruning strategies

(b) Experimental results for RNN models

Fig. 4. Experimental results according to the training and test data ratio.

CRediT authorship contribution statement

Chanhee Park: Conceptualization, Data curation, Formal analysis, Methodology, Validation, Writing - original draft, Writing - review & editing, Software. **Jinuk Park:** Conceptualization, Writing - original draft, Investigation, Validation. **Sanghyun Park:** Conceptualization, Funding acquisition, Project administration, Supervision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Next Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT (NRF 2015M3C4A7065522).

References

Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., & Salakoski, T. (2008). All-paths graph kernel for protein-protein interaction extraction with evaluation of cross-corpus learning. *BMC Bioinformatics, 9*(suppl. 11). https://doi.org/10.1186/1471-2105-9-52.

Bahdanau, D., Cho, K., & Bengio, Y. (2015). Neural machine translation by jointly learning to align and translate. Proceedings of the ICLR 2015 – International Conference on Learning Representations. Retrieved from http://arxiv.org/abs/1409.0473.

Bastings, J., Titov, I., Aziz, W., Marcheggiani, D. & Simaan, K. (2017). Graph Convolutional Encoders for Syntax-aware Neural Machine Translation. Proceedings of the EMNLP 2017 – Conference on Empirical Methods in Natural Language Processing 1957–1967. https://doi.org/10.18653/v1/d17-1209

Björne, J., Kaewphan, S. & Salakoski, T. (2013). UTurku: Drug named entity recognition and drug-drug interaction extraction using SVM classification and domain knowledge. Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (∗SEM), Volume 2: 7th International Workshop on Semantic Evaluation (SemEval 2013) (pp. 651–659).

Chowdhury, M. F. M. & Lavelli, A. (2013a). Exploiting the scope of negations and heterogeneous features for relation extraction: A case study for drug-drug interaction extraction. Proceedings of the NAACL HLT 2013 – 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (pp. 765–771).

Chowdhury, M. F. M. Lavelli, A. (2013b). FBK-irst: A multi-phase kernel based approach for drug-drug interaction detection and classification that exploits linguistic information. Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (∗SEM), Volume 2: 7th International Workshop on Semantic Evaluation (SemEval 2013) (pp. 351–355).

Duvenaud, D., Maclaurin, D., Aguilera-Iparraguirre, J., Gómez-Bombarelli, R., Hirzel, T., Aspuru-Guzik, A. & Adams, R. P. (2015). Convolutional networks on graphs for learning molecular fingerprints. Proceedings of the NIPS 2015 – Advances in Neural Information Processing Systems (pp. 2224–2232).

Fundel, K., Küffner, R., & Zimmer, R. (2007). RelEx – relation extraction using dependency parse trees. *Bioinformatics, 23*(3), 365–371. https://doi.org/10.1093/bioinformatics/btl616.

Giuliano, C., Lavelli, A., Pighin, D. & Romano, L. (2007). FBK-IRST: Kernel methods for semantic relation extraction. Proceedings of the ACL 2007 – SemEval 2007 – the 4th International Workshop on Semantic Evaluations (pp. 141–144).

Giuliano, C., Lavelli, A., & Romano, L. (2006). Exploiting shallow linguistic information for relation extraction from biomedical literature. Proceedings of the EACL 2006 – 11th Conference of the European Chapter of the Association for Computational Linguistics (pp. 401–408).

Gori, M., Monfardini, G. & Scarselli, F. (2005). A new model for earning in graph domains. Proceedings of the 2005 IEEE International Joint Conference on Neural Networks (pp. 729–734). https://doi.org/10.1109/IJCNN.2005.1555942.

Huang, D., Jiang, Z., Zou, L., & Li, L. (2017). Drug–drug interaction extraction from biomedical literature using support vector machine and long short term memory networks. *Information Sciences, 415*, 100–109. https://doi.org/10.1016/j.ins.2017.06.021.

Kim, S., Liu, H., Yeganova, L., & Wilbur, W. J. (2015). Extracting drug-drug interactions from literature using a rich feature-based linear kernel approach. *Journal of Biomedical Informatics, 55*, 23–30. https://doi.org/10.1016/j.jbi.2015.03.002.

Kipf, T. N. & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. Proceedings of the ICLR 2017 – International Conference on Learning Representations. Retrieved from http://arxiv.org/abs/1609.02907.

Lee, K., He, L., Lewis, M. & Zettlemoyer, L. (2018). End-to-end neural coreference resolution. Proceedings of the EMNLP 2018 – Conference on Empirical Methods in Natural Language Processing (pp. 188–197). https://doi.org/10.18653/v1/d17-1018.

Li, Y., Zemel, R., Brockschmidt, M. & Tarlow, D. (2016). Gated graph sequence neural networks. Proceedings of the ICLR 2016 - International Conference on Learning Representations.

Lin, J., Sun, X., Ma, S. & Su, Q. (2018). Global encoding for abstractive summarization. Proceedings of the ACL 2018 – 56th Annual Meeting of the Association for Computational Linguistics (pp. 163–169).

Luong, M. T., Pham, H. & Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. Proceedings of the EMNLP 2015 – Conference on Empirical Methods in Natural Language Processing (pp. 1412–1421).

Liu, S., Tang, B., Chen, Q., & Wang, X. (2016a). Drug-drug interaction extraction via convolutional neural networks. *Computational and Mathematical Methods in Medicine*. https://doi.org/10.1155/2016/6918381.

Liu, S., Tang, B., Chen, Q., & Wang, X. (2016b). Dependency-based convolutional neural network for drug-drug interaction extraction. *Proceedings of the BIBM 2016 - IEEE International Conference on Bioinformatics and Biomedicine*, 1074–1080. https://doi.org/10.1109/BIBM.2016.7822671.

Manessi, F., Rozza, A., & Manzo, M. (2020). Dynamic graph convolutional networks. *Pattern Recognition, 97*. https://doi.org/10.1016/j.patcog.2019.107000.

Maratea, A., Petrosino, A., & Manzo, M. (2014). Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences, 257*, 331–341. https://doi.org/10.1016/j.ins.2013.04.016.

Miwa, M. & Bansal, M. (2016). End-to-end relation extraction using LSTMs on sequences and tree structures. Proceedings of the ACL 2016 – 54th Annual Meeting of the Association for Computational Linguistics (pp. 1105–1116).

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T. & Ananiadou, S. (2013). Distributional semantics resources for biomedical text processing. Proceedings of the LBM 2013 – Languages in Biology and Medicine Conference (pp. 39–44).

Quan, C., Hua, L., Sun, X., & Bai, W. (2016). Multichannel convolutional neural network for biological relation extraction. *BioMed Research International*. https://doi.org/10.1155/2016/1850404.

Raihani, A., & Laachfoubi, N. (2016). Extracting drug-drug interactions from biomedical text using a feature-based kernel approach. *Journal of Theoretical and Applied Information Technology, 92*, 109–120.

Sahu, S. K., & Anand, A. (2018). Drug-drug interaction extraction from biomedical texts using long short-term memory network. *Journal of Biomedical Informatics, 86*, 15–24. https://doi.org/10.1016/j.jbi.2018.08.005.

Santoro, A., Raposo, D., Barrett, D. G. T., Malinowski, M., Pascanu, R., Battaglia, P. & Lillicrap, T. (2017). A simple neural network module for relational reasoning. Proceedings of the NIPS 2017 – Advances in Neural Information Processing Systems (pp. 4974–4983).

Scarselli, F., Gori, M., Tsoi, A. C., Hagenbuchner, M., & Monfardini, G. (2009). The graph neural network model. *IEEE Transactions on Neural Networks, 20*(1), 61–80. https://doi.org/10.1109/TNN.2008.2005605.

Segura-Bedmar, I., Martínez, P. & Herrero-Zazo, M. (2013). SemEval-2013 task 9: Extraction of drug-drug interactions from biomedical texts (DDIExtraction 2013). Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (∗SEM), Volume 2: 7th International Workshop on Semantic Evaluation (SemEval 2013) (pp. 341–350).

Seo, M., Kembhavi, A., Farhadi, A. & Hajishirzi, H. (2017). Bidirectional attention flow for machine comprehension. Proceedings of the ICLR 2017 – International Conference on Learning Representations. https://doi.org/10.18653/v1/P17-1055

Seo, Y., Defferrard, M., Vandergheynst, P. & Bresson, X. (2018). Structured sequence modeling with graph convolutional recurrent networks. Proceedings of the ICONIP 2018 – International Conference on Neural Information Processing (pp. 362–373). https://doi.org/10.1007/978-3-030-04167-0_33.

Sun, X., Dong, K., Ma, L., Sutcliffe, R., He, F., Chen, S., & Feng, J. (2019). Drug-drug interaction extraction via recurrent hybrid convolutional neural networks with an improved focal loss. *Entropy, 21*(37). https://doi.org/10.3390/e21010037.

Thomas, P., Neves, M., Rocktäschel, T. & Leser, U. (2013). WBI-DDI: drug-drug interaction extraction using majority voting. Proceedings of the 2nd Joint Conference on Lexical and Computational Semantics (∗SEM), Volume 2: 7th International Workshop on Semantic Evaluation (SemEval 2013) (pp. 628–635).

Tsuruoka, Y., Tateishi, Y., Kim, J. D., Ohta, T., McNaught, J., Ananiadou, S. & Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. Proceedings of the PCI 2005 – the 10th Panhellenic Conference in Informatics (pp. 382–392). https://doi.org/10.1007/11573036_36.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... Polosukhin, I. (2017). Attention is all you need. Proceedings of the NIPS 2017 – Advances in Neural Information Processing Systems.

Wishart, D. S., Feunang, Y. D., Guo, A. C., Lo, E. J., Marcu, A., Grant, J. R., ... Wilson, M. (2018). DrugBank 5.0: A major update to the DrugBank database for 2018. *Nucleic Acids Research, 46*(D1), D1074–D1082. https://doi.org/10.1093/nar/gkx1037.

Xu, K., Feng, Y., Huang, S., & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the EMNLP 2015 - Conference on Empirical Methods in Natural Language Processing* (pp. 536–540).

Xu, K., Feng, Y., Huang, S. & Zhao, D. (2015). Semantic relation classification via convolutional neural networks with simple negative sampling. Proceedings of the EMNLP 2015 – Conference on Empirical Methods in Natural Language Processing (pp. 536–540).

Yi, Z., Li, S., Yu, J., Tan, Y., Wu, Q., Yuan, H. & Wang, T. (2017). Drug-drug interaction extraction via recurrent neural network with multiple attention layers. Proceedings of the ADMA 2017 – Advanced Data Mining and Applications, Lecture Notes in Computer Science (Vol. 10604, pp. 554–566). https://doi.org/10.1007/978-3-319-69179-4_39.

Yu, A. W., Dohan, D., Luong, M. -T., Zhao, R., Chen, K., Norouzi, M. & Le, Q. V. (2018). QANet: Combining local convolution with global self-attention for reading comprehension. Proceedings of the ICLR 2018 – International Conference on Learning Representations. Retrieved from http://arxiv.org/abs/1804.09541

Zeng, D., Liu, K., Lai, S., Zhou, G. & Zhao, J. (2014). Relation classification via convolutional deep neural network. Proceedings of the COLING 2014 – 25th International Conference on Computational Linguistics, Proceedings of COLING 2014: Technical Papers (pp. 2335–2344).

Zhang, Yijia, Lin, H., Yang, Z., Wang, J., & Li, Y. (2012). A single kernel-based approach to extract drug-drug interactions from biomedical literature. *PLoS One, 7*(11). https://doi.org/10.1371/journal.pone.0048901.

Zhang, Yijia, Zheng, W., Lin, H., Wang, J., Yang, Z., & Dumontier, M. (2018a). Drug-drug interaction extraction via hierarchical RNNs on sequence and shortest dependency paths. *Bioinformatics, 34*(5), 828–835. https://doi.org/10.1093/bioinformatics/btx659.

Zhang, Yuhao, Qi, P. & Manning, C. D. (2018b). Graph convolution over pruned dependency trees Improves relation extraction. Proceedings of the EMNLP 2018 – Conference on Empirical Methods in Natural Language Processing (pp. 2205–2215). https://doi.org/10.18653/v1/d18-1244.

Zhao, Z., Yang, Z., Luo, L., Lin, H., & Wang, J. (2016). Drug drug interaction extraction from biomedical literature using syntax convolutional neural network. *Bioinformatics, 33*(22), 3444–3453. https://doi.org/10.1093/bioinformatics/btw486.

Zhou, D., Miao, L., & He, Y. (2018). Position-aware deep multi-task learning for drug–drug interaction extraction. *Artificial Intelligence in Medicine, 87*, 1–8. https://doi.org/10.1016/j.artmed.2018.03.001.