Proceedings of Machine Learning Research 129:129–144, 2020 ACML 2020

Bidirectional Dependency-Guided Attention for Relation Extraction

Xingchen Deng DENGXC@BUPT.EDU.CN
Lei Zhang ZLEI@BUPT.EDU.CN
School of Computer Science, Beijing University of Posts and Telecommunications

Yixing Fan FANYIXING@ICT.AC.CN
Long Bai BAILONG18B@ICT.AC.CN
JiaFeng Guo GUOJIAFENG@ICT.AC.CN
Institute of Computing Technology, Chinese Academy of Sciences

Pengfei Wang WANGPENGFEI@BUPT.EDU.CN
School of Computer Science, Beijing University of Posts and Telecommunications

Editors: Sinno Jialin Pan and Masashi Sugiyama

Abstract

The dependency relation between words in the sentence is critical for the relation extraction. Existing methods often utilize the dependencies accompanied with various pruning strategies, thus suffer from the loss of detailed semantic information. In order to exploit dependency structure more effectively, we propose a novel bidirectional dependency-guided attention model. The main idea is to use a top-down attention as well as a bottom-up attention to fully capture the dependencies from different granularity. Specifically, the bottom-up attention aims to model the local semantics from the subtree of each node, while the top-down attention is to model the global semantics from the ancestor nodes. Moreover, we employ a label embedding component to attend the contextual features, which are extracted by the dependency-guided attention. Overall, the proposed model is fully attention-based which make it easy for parallel computing. Experiment results on TACRED dataset and SemEval 2010 Task 8 dataset show that our model outperforms existing dependency based models as well as the powerful pretraining model. Moreover, the proposed model achieves the state-of-the-art performance on TACRED dataset.

Keywords: relaion extraction, dependency tree, attention mechanism

1. Introduction

Reading text to identify and extract relations between entities has been a long standing goal in natural language processing. Take the following sentence as an example: *"[Markus Andreas]$_{e1}$, arrested for murdering an engineer in Polish, was a former [Austrian]$_{e2}$ lawmaker ."* The relation between entity "*Markus Andreas*" and entity "*Austria*" is "*per:origin*". High-quality relation extraction could offer useful information for many applications, such as question answering, information extraction, the construction and completion of knowledge base.

The head-dependent relations parsed by dependency grammar express the semantic relationships between different constituents of the sentence which makes them directly useful for many applications such as coreference resolution and information extraction. Thus,

dependency parsing tree of the sentence has been used as a strong feature from some traditional method [Zelenko et al. (2003); KAMBHATLA (2004)] , because model can capture long-range dependency relation information and reduce the noise of irrelevant words with the help of dependency information. However, traditional models face the challenge of sparse feature spaces and are brittle to lexical variations.

Recent years, to solve above problems, deep learning methods are widely applied to this task and several work show that incorporating dependency trees into deep neural models significantly improves the performance. However, to avoid noise of irrelevant context, most models adopt pruning strategies. For instance, Miwa and Bansal (2016) and Xu et al. (2015) found it is quite effective applying LSTM over the lowest common ancestor (LCA) subtree of entities and even only the shortest dependency path of entities. Nevertheless, due to the limit of above models' pruning strategies and their structures, they suffer from serious information loss and some of them are computationally inefficient. To overcome these shortcomings, some gragh based models have been proposed: Zhang et al. (2018) proposes contextualized graph convolutional networks (C-GCN) model over a pruned tree which use a path-centric pruning strategy; Guo et al. (2019) proposed an attetntion guided GCN model (AGGCN) which is a graph attention model based on full dependency tree. But the pruning strategy used in C-GCN to alleviate information loss is still aggressive and harmful to dependency structure because a certain K hop path-centric pruning won't be suitable for all cases. For example, in the sentence showed in figure 1, using such pruning strategy, the sentence will lose the information: "*murdering an engineer in Polish*" , which imply the meaning that this man is from other place. Thus the model is more likely to predict this sample as "per:countries of residence" for "person - country" entity pair due to unbalanced data. Moreover, the C-GCN and AGGCN model both use a sparse adjacency matrix, node on the tree can only interact with its parent node and child nodes. Multiple layers stacking is needed to interact with more and farther nodes, which may bring information confusion and information loss when stacking.

In fact, on the dependency tree, a subtree for any none-leaf node represents a context constituent (e.g, a noun/verb phrase, a subordinate clause) and this none-leaf node is the central organizing word (head node) of the constituent, other words in the constituent are dependent nodes. Thus, for any none-leaf node on the dependency tree, the information flow collected from bottom-up on the subtree of the node expresses detailed local semantics of a smaller constituent. The top-down information flow collected from ancestors dependency path of the node indicates global semantics for a larger and more specific constituent where the gathered local semantics of the node will blend in. In other words, for the bottom-up direction, the node gather the local information as a head node, for the top-down direction, the node gather the global information from larger scope as a dependent node. Therefore, using the pruned dependency tree without direction may do damage to semantic aggregation and degrade the performance.

In this paper, we propose a novel dependency based model that are also fully attention-based to overcome above shortcomings, we call it bidirectional dependency guided attention(Bi-DGA) model. The Bi-DGA is consisted of bottom-up attention and top-down attention. Guided by dependency, bottom-up attention allows a node interacting with all nodes of its subtree on dependency tree(all descendant nodes of current node on dependency tree), then collects the information flow from bottom-up. Likely, top-down attention receive the

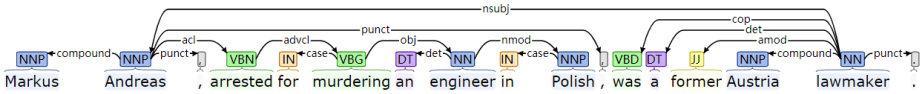


Figure 1: The dependency parse of sentence "Markus Andreas, arrested for murdering an engineer in Polish, was a former Austria lawmaker. " This visualization result comes from stanford corenlp.

information from top-down ancestors dependency path by an attention operation with all ancestor nodes of current node on the dependency tree. Inspired by transformer [Vaswani et al. (2017)] which use a lower triangular mask matrix to make sure the word in the sentence only interact with words before itself to encode the sentence order. Our model encode the tree structure by bottom-up attention and top-down attention with bottom-up masks and top-down masks. In this way, we build information flows for two different directions. Meanwhile, a single node is able to interact with more nodes related on semantic. Beyond that, instead of rule-based pruning strategies, we keep the whole tree and use a hop embedding which indicate the hops to dependency path of two entities on the LCA tree. Moreover, extra knowledge is helpful for improving the performance of relation extraction, especially under the circumstances of unbalanced data. In our model, we encode the word representations using ALBERT, a pretrained language model [Lan et al. (2020)] that has significantly fewer parameters than a BERT architecture [Devlin et al. (2019)] , to obtain word representations with rich and context-related lexical information. In addition, with the idea that the same label may contain similar context information as clues for classification, we employ a label embedding component and label-to-context attention to attend the label-related contextual features for each label without external human-designed knowledge features. Then, we transform the classification problem into a matching problem with hinge loss. Through these ways, our model is able to reach a high score with more balanced precision and recall.

Our contributions are as follows: (1) We propose a novel bidirectional dependency guided attention mechanism, which allows it to encode dependency tree structure to capture long-range syntactic relation information without using an aggressive pruning strategy. We use bottom-up attention and top-down attention to gather local and global semantics respectively, based on the character of dependency tree. Thus, the Bi-DGA provide a efficient way to encode tree structure with attention mechanism. (2) We use label-to-context attention to extract label-related context features as classification clues and transform the classification problem into a matching problem to further improve the performance. (3) We test our model on SemEval 2010 Task 8 dataset and the larger TACRED dataset and our performance on both datasets surpassing several competitive baselines. The state-of-the-art performance is achieved on TACRED dataset.

2. Related Work

Traditional relation extraction methods based on human-design features [KAMBHATLA (2004); Hendrickx et al. (2009)] or kernels [Bunescu and Mooney (2005); Plank and Moschitti (2015)] have the disadvantage of time-consuming and poor generalization due to the low coverage of different training datasets. Recent years, the focus of research on supervised relation extraction methods has shifted to neural models. Zeng et al. (2014) first apply CNNs to relation extraction. Nguyen and Grishman (2015) improve the results by introducing multiple convolution windows sizes. Zhang and Wang (2015) first apply Recurrent Neural Network(RNN) to this task and get competitive performance to CNNs models. Vu et al. (2016) showed that combining a CNN with a RNN through a voting scheme can further improve performance. [Wang et al. (2016); Peng et al. (2016); Zhang et al. (2017)] proposed to use attention mechanisms to capture important information over RNN and CNN architectures for this task and achieve high performance. Li et al. (2019) implements an attention mechanism which incorporates prior knowledge from external human-designed lexical resources of labels and reach a competitive results.

Dependency parsing tree of the sentence has been used as a strong feature from some traditional method [Zelenko et al. (2003); KAMBHATLA (2004)]. Because model can capture long-range syntactic relation information and reduce the influence of irrelevant noise with the help of dependency information. Several work show that incorporating dependency trees into neural models is also helpful. Xu et al. (2015) generalized the idea of dependency path kernels by applying a LSTM network over the shortest dependency path between entities. Miwa and Bansal (2016) applied a Tree-LSTM, a generalized form of LSTM over dependency trees, in a joint entity and relation extraction setting. They use the shortest dependency path on LCA tree of the two entities to extract features effectively. Zhang et al. (2018) use graph convolutional neural network and a path-centric pruning strategy of dependency trees to keep relative information and further improve the performance. Guo et al. (2019) proposed an attetntion guided GCN model (AGGCN) which is a graph attention model based on full dependency tree which uses graph attention to focus on important information and further improves the performance. But the LSTM model using dependency information is hard to encode tree structure with multiple children structure and suffer from low computational efficiency. The GCN and AGGCN models improved these two problems. However, they both use the tree as an undirected graph which is still harmful to the tree structure and suffer from the problem of sparse adjacency matrix.

Recent two years, unsupervised pre-training language models have shown to be a very effective and improved performance on various natural language processing tasks. The Generative Pre-trained Transformer (OpenAI GPT) [Radford et al. (2018)], a left-to-right transformer based language model pretrained on large corpus, achieved significant results on many sentence level tasks. BERT [Devlin et al. (2019)] further improved the performance on lots of NLP tasks by using masked language models to enable pretrained deep bidirectional representations. By incorporating factorized embedding parameterization and cross-layer parameter sharing, ALBERT [Lan et al. (2020)], a lite BERT architecture, has significantly fewer parameters than a traditional BERT architecture. Joshi et al. (2019) extends BERT by masking contiguous random spans and training the span boundary representations to predict the entire content of the masked span, without relying on the individual token

representations within it. Extra knowledge is helpful for improving the performance of relation extraction, especially under the circumstances of unbalanced data. Pretrained language model which trained on large external corpus is able to provides rich co-occurrence and semantic information. Researchers also tried to improve the performance of relation classification via fine-tuning on pretrained language model [Alt et al. (2019); Wu and He (2019); Joshi et al. (2019)]. In our work, for fewer parameters and better performance, we use ALBERT to generate word representation.

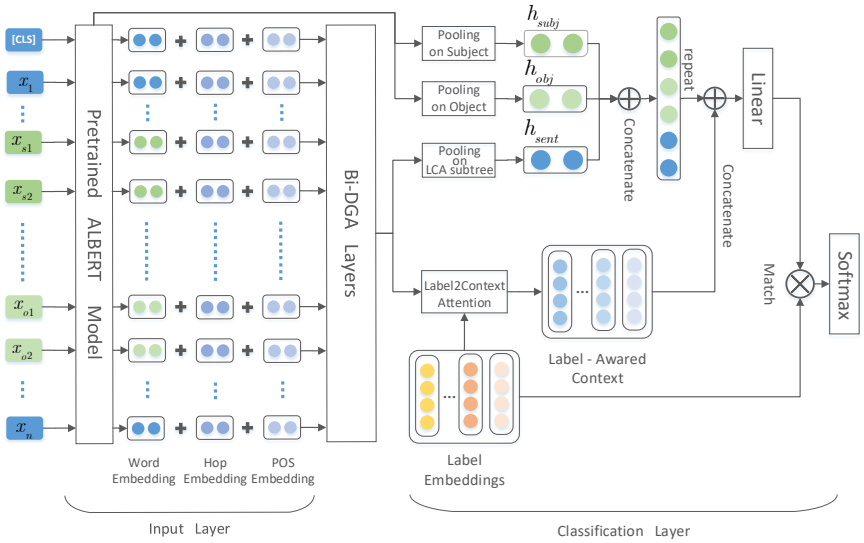


Figure 2: Overall architecture of the Bi-DGA model

3. Methodology

For supervised relation extraction, the task can be formalized as follows: Let $X = [x_1, x_2, \ldots, x_n]$ denote a sentence, where x_i is the i-th word token. A subject entity and an object entity are identified and correspond to two spans in the sentence: $X_s = [x_{s_1}, x_{s_2}, \ldots, x_{s_n}]$ and $X_o = [x_{o_1}, x_{o_2}, \ldots, x_{o_n}]$. Given X, X_s and X_o, the goal of relation classification is to predict a relation $r \in R$ (a predefined relation set) that holds between the entities or no relation otherwise.

The overall architecture of our model is shown in figure 2, which is consisted of the following three modules: (1) Input layer: tokens are fed into pretrained ALBERT model[1] to get embedding for each word. Then the word embedding are concatenated with POS tag embedding and hop embedding to form the input feature h_0 of Bi-DGA layer. (2) Bi-DGA layers: h_0 are encoded on the dependency tree for layers to get the structure guided

1. ALBERT may tokenize a word to several subwords, to keep the dependency structure, we tried: 1) use the first subword as node on dependency tree and mask other subwords; 2) treat other subwords as child nodes of first subword; We found that the second approach reach a better performance.

features. (3) Classification layer: the outputs of ALBERT are pooling on subject positions, object positions and LCA subtrees respectively to get features h_{subj} , h_{obj}, and h_{sent}. And we use label-to-context attention to obtain a label-related context feature for each label. Then we concatenate these features together and feed them into a linear layer. After that, the features will be matched with label embedding to get the final scores.

3.1. Input Representation

For relation extraction, the datasets are usually unbalanced, with the traditional static word vectors [Mikolov et al. (2013); Pennington et al. (2014)] , fewer training samples of a label lead to the lack of enough semantic information for identifying this label. BERT [Devlin et al. (2019)] is a deep bidirectional transformer model pretrained on BooksCorpus and Wikipedia. Thus, BERT can generate contextualized word representations and provide richer lexical and semantic information than traditional static word vectors, which is helpful for improving the performance. For fewer parameters and comparative performance, we use ALBERT, a lite BERT architecture, as our word representations encoder.

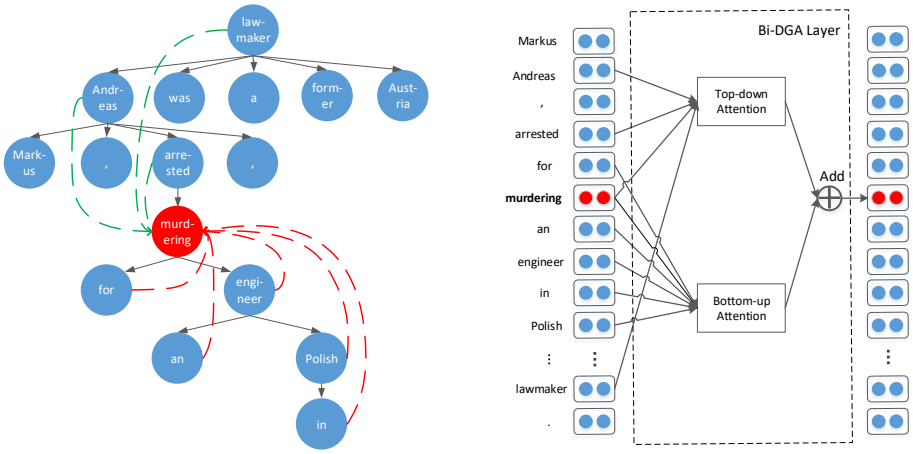
For each input sentence, a '[CLS]' token is appended to the beginning of token sequence. Input representation of each token for ALBERT is constructed by the summation of the corresponding token, segment and position embeddings. We denote the hidden state output from ALBERT is $h_w \in \mathbb{R}^{n \times d_w}$, where n is the sequence length of input sentence (including '[CLS]') and d_w is the size of hidden state output of ALBERT. And h_{wi} is the hidden state feature of token i.

Previous work [Xu et al. (2015); Miwa and Bansal (2016); Zhang et al. (2018)] indicate that extracting features on shortest dependency path on LCA subtree of two entities is effective for most of cases. However, such a strategy is too aggressive which may neglect important information for some cases as demonstrated in Section 1. Besides, a K hop pruning strategy used in C-GCN model [Zhang et al. (2018)] leads to the same problem because a certain K distance pruning will not be suitable for all cases. For further improving performance, we maintain the whole dependency tree and use the hop embedding as auxiliary information for attention. For each word, we calculate its hop-distance K away from dependency path, if K is equal to 0, indicates that the word is unreachable, such as a padding word. If K is equal to 1, indicates that the word is on the dependency path. Otherwise, K indicate that the node is K-1 hop away from the dependency path. For nodes are far away from dependency path (hops > 7), we set K to 7. Then the hop-distance K of each word will be transformed into a vector by looking up the embedding matrix $V_{hop} \in \mathbb{R}^{d_{hop} \times K_{max}}$, where d_{hop} is the dimension of hop embedding vector and K_{max} is the maximum of hop K.

Follow previous works, we concatenate the word embedding produced by ALBERT, the POS tag embeddings as well as the hop embedding proposed by us to generate the input features $h_0 \in \mathbb{R}^{n \times d_s}$ for Bi-DGA layers. Here, $d_s = d_w + d_{hop} + d_{pos}$, d_{pos} is the dimension of POS tag embedding.

3.2. Bidirectional Dependency Guided Attention Mechanism

Attention mechanism has been successfully applied in relation extraction models. However, most of these models simply use attention mechanism to extract relative features in



(*a*) The information flow on dependency tree

(*b*) Bi-DGA layer

Figure 3: An example of how the Bi-DGA generates the features of the token "murdering" for next layer .The (a) shows the information flow from top-down and bottom-up. The green dashed lines represent the top-down information flow, the red dashed lines represent the bottom-up information flow. The (b) shows the computation process of Bi-DGA layer for the token "murdering".

decoder stage. They still use CNNs or LSTM as their encoder that are difficult to encode the entire sentence over dependency tree.

Here we propose the bidirectional dependency guided attention. An example describes how does it work is shown in figure 3. Instead of an attention operation for all nodes from two directions, the bidirectional dependency guided attention is consisted of bottom-up attention and top-down attention. Bottom-up attention is an attention operation for all descendant nodes of current node and itself. Top-down attention is an attention operation with all ancestor nodes on the dependency tree. On the basis of the character of the dependency tree, we hold the opinion that the node gather information from two directions as head node and dependent node respectively, then obtain semantics for different semantic scope. We believe the information flow from two directions are equally important and they ought to be encoded respectively before combining them together. The importance of nodes in one direction for current node is not supposed to be influenced by nodes from another direction during attention stage which may result in information confusion between two information flows and damage to the semantic aggregation. Thus, encoding the sentence with directional dependency tree is also important. Our experiments in Section 4 demonstrate that a bidirectional dependency attention can indeed improve the performance.

In an L-layer Bi-DGA, if we denote by $h^{(l-1)} \in \mathbb{R}^{n \times d_{(l-1)}}$, $h^l \in \mathbb{R}^{n \times d_l}$ the input features and the output features at the l-th layer, a bottom-up attention operation can be computed as below: First, like most commonly used attention [Vaswani et al. (2017)], we calculate Q, K, V and get the scores between all elements through Q and K.

$$Q^l = h^{(l-1)}{W_q}^l + {b_q}^l \quad , \quad K^l = h^{(l-1)}{W_k}^l + {b_k}^l \quad , \quad V^l = h^{(l-1)}{W_v}^l + {b_v}^l \tag{1}$$

$$S^l = (Q^l {K^l}^T)/\sqrt{d_k} \tag{2}$$

Where ${W_q}^l \in \mathbb{R}^{d_{(l-1)} \times d_k}$, ${W_k}^l \in \mathbb{R}^{d_{(l-1)} \times d_k}$ and ${W_v}^l \in \mathbb{R}^{d_{(l-1)} \times d_v}$ are linear transformation matrices, ${b_q}^l$,${b_k}^l$,${b_v}^l$ and ${b_o}^l$ are bias terms, d_k is the dimension of Q and K, d_v is the dimension of V.

For each node on the dependency tree, we want it to interact with all its descendant nodes. We implement it by using a bottom-up mask matrix $Mask_{bottom}$.

$$Mask_{bottom}(i,j) = \begin{cases} 0 & \text{if} \quad j \in \mathcal{D}(i) \quad \text{or} \quad j = i \\ \\ -\infty & \text{else} \end{cases} \tag{3}$$

$${S_m}^l = Softmax(S^l + Mask_{bottom}) \tag{4}$$

Where $\mathcal{D}(i)$ is the descendant nodes of node i. Then we can calculate the output of bottom-up attention ${h_{bottom}}^l$ with ${S_m}^l$:

$${h_{bottom}}^l = g((V^l {S_m}^l){W_o}^l + {b_o}^l) \tag{5}$$

Where ${W_o}^l \in \mathbb{R}^{d_v \times d_l}$, ${b_o}^l$ is bias term, d_l is the hidden size of the bottom-up attention output features for layer l. $g(\cdot)$ is gelu activation [Hendrycks and Gimpel (2016)].

The output of top-down attention ${h_{top}}^l$ can be obtained in the same way with top-down mask matrix and different transformation matrices and bias terms for calculating Q,K,V and ${h_{top}}^l$. And the $h^l \in \mathbb{R}^{n \times d_l}$ is the sum of ${h_{bottom}}^l$ and ${h_{top}}^l$:

$$h^l = {h_{bottom}}^l + {h_{top}}^l \tag{6}$$

3.3. Classification Layer

Concatenating the sentence and the entity representations before classifying has been shown effective in previous work [Santoro et al. (2017);Lee et al. (2017);Zhang et al. (2018)]. Therefore, we obtain the subject and object entity representations from the output hidden features h_w of ALBERT by pooling:

$$h_{subj} = pool(h_{w[s_1:s_n]}) \qquad h_{obj} = pool(h_{w[o_1:o_n]}) \tag{7}$$

Where $h_{subj} \in \mathbb{R}^{d_w}$, $h_{obj} \in \mathbb{R}^{d_w}$ and $pool(\cdot) : \mathbb{R}^{n \times d} \to \mathbb{R}^d$ is a max pooling function.

We get the sentence representations h_{sent} by pooling the output features of Bi-DGA layers h^L on the complete common ancestor subtree (including common ancestor node and all its descendant nodes, denote this nodes collection by $\mathcal{T}$) of two entities. Thus, the h_{sent} contain the relation semantics between two entities.

$$h_{sent} = pool({h^L}_{[t \in \mathcal{T}]}) \tag{8}$$

Previous work [Wang et al. (2016);dos Santos et al. (2015)] shows that it is effective that using label information to attend the contextual features. Similarly, we think samples with the same label may contain similar context information which could be used as a clue to help classification, we also introduce a relation label embeddings matrix $V_R \in \mathbb{R}^{m \times d_r}$, where m is the number of labels and d_r is the dimension. To get label-related context features for each label, we implement a label-to-context attention as follows:

$$Q^{'} = h^L W_{q^{'}} + b_{q^{'}} \quad , \quad K^{'} = V_R W_{k^{'}} + b_{k^{'}} \quad , \quad V^{'} = h^L W_{v^{'}} + b_{v^{'}} \tag{9}$$

We use the output features h^L of Bi-DGA layers to generate $Q^{'} \in \mathbb{R}^{n \times d_k}$ and $V^{'} \in \mathbb{R}^{m \times d_k}$, and use label embeddings V_R to generate $K^{'} \in \mathbb{R}^{n \times d_k}$. Then we obtain a matrix $S^{'} \in \mathbb{R}^{n \times m}$ with the scores of similarity between label embeddings and the token features .

$$S^{'} = (\frac{Q^{'} K^{'T}}{\sqrt{d_k}}) \tag{10}$$

To normalize the importance scores of tokens under single label, we transpose the $S^{'}$, and do softmax operation at the last dimension:

$$S^{''} = Softmax(S^{'T}) \tag{11}$$

With $S^{''}$, we can calculate the label related context features for each label:

$$H_R = g((S^{''} V^{\prime}) W^{'} + b^{'}) \tag{12}$$

Where $W^{'} \in \mathbb{R}^{d_v \times d_h}$, $H_R \in \mathbb{R}^{m \times d_h}$ and d_h is a hidden size. So far, we have obtained the entity features : h_{subj} and h_{obj}, the sentence representations with relation semantics : h_{sent}, and the label related context features: H_R . Then we combine them together by concatenating. First, we concatenate h_{subj}, h_{obj} and h_{sent} at the last dimension and repeat it for m times to generate the feature $H_C \in \mathbb{R}^{m \times d_c}$ ($d_c = d_w + d_w + d_L$). Then we obtain the final representations H_{final} through concatenating H_C with H_R and feeding them into a feed-forward neural network (FFNN):

$$H_{final} = FFNN([H_C; H_R]) \tag{13}$$

Not that for the convenience of mathching operation, after $FFNN(\cdot)$, the H_{final} must have the same dimension with label embeddings . The H_{final} is then matched with label embeddings to obtain a score distribution $\tilde{\mathbf{y}}$ over relations for the target entity pair. For the score of label i:

$$\tilde{y_i} = H_{final_i} \cdot V_{Ri} \tag{14}$$

Where $\cdot$ is dot production, H_{final_i} is the i-th feature of H_{final} and V_{Ri} is the embedding of i-th label. Then we can get the predicted label $\hat{y}$ of the target entity pair :

$$\hat{y} = \arg\max_i(\tilde{y_i}) \tag{15}$$

3.4. Loss Fuction

For the loss function, we use the hinge loss which is widely used in matching problems. For an instance with label $\hat{y}_i$, the loss is calculated as below:

$$\mathcal{L} = \sum_{\hat{y}_j \in R, \hat{y}_j \neq \hat{y}_i} MAX(0, \tilde{y}_j - \tilde{y}_i + 1) \tag{16}$$

we also tried to use cross entropy as loss function, which result in a high recall and a low precision, and the f1 score using cross entropy is slightly lower than that using hinge loss. We suppose that's because the hinge loss is able to expand the distance in vector space between categories to distinguish similar labels.

4. Experiment

4.1. Dataset

We conduct experiments on two relation extraction datasets :

(1) TACRED: TACRED is a large-scale relation extraction dataset with over 106K sentences with mention pairs introduced in [Zhang et al. (2017)]. The entity pairs are annotated by subject or object. It represents 41 relation types and a *"no_relation"* class when the mention pair does not have a relation between them within these categories. The dataset is unbalanced with 79.5% *"no_relation"* samples, which makes it difficult for models to extract relations for positive samples. Follow previous work, a "entity mask" strategy is used to replace subject (or object) entity with "<NER Type>-SUBJ" (or " <NER Type>-OBJ ") and report micro-averaged F_1 score on this dataset. We select our best model based on the median validation F1 score over 5 independent runs and report its performance on the test set.

(2) SemEval 2010 Task 8: The SemEval 2010 dataset is much smaller and simpler than TACRED with 8000 training samples and 2717 testing samples. It contains 9 directed semantic relations types, such as "Entity-Origin(e1,e2)", "Entity-Origin(e2,e1)" . And the *Other* relation indicates that there is no relation between two entities. Therefore, there are 19 relations classes in total. We use this dataset to evaluate the generalization ability of our proposed model. The evaluation metric for this dataset is official macro averaged F_1 score.

4.2. Experimental Settings

We use the ALBERT-V2 to generate word vector representations. The parameters of pos tag embedding, hop embedding, label embedding are randomly initialized, and the d_{pos}, d_{hop} are set to be 30. The d_k and d_v for all attention operations are 32. For TACRED dataset, the hidden size d_l for all Bi-DGA layers l is 256, the dimension of label embedding and the hidden size d_h for label-to-context attention are 128. For SemEval 2010 with a much smaller data size, we half the d_l and d_h and only use ALBERT-base as pretrained encoder. We find that 2-layer stacked Bi-DGA layers work best for both two dataset. We minimize the hinge loss using *AdamW* optimizer [Loshchilov and Hutter (2019)] with a learning rate of 8e-6 for TACRED with ALBERT-base and 5e-6 for ALBERT-large as well

Model	P	R	F_1
Sequence Model			
PA-LSTM † [Zhang et al. (2017)]	65.7	64.5	65.1
Knwl-attn † [Li et al. (2019)]	70.1	66.0	67.9
Dependency Based Model			
LR‡ [Zhang et al. (2017)]	**73.5**	49.9	59.4
SDP-LSTM ‡ [Xu et al. (2015)	66.3	52.7	58.7
Tree-LSTM ‡ [Tai et al. (2015)]	66.0	59.2	62.4
C-GCN † [Zhang et al. (2018)]	69.9	63.3	66.4
C-AGGCN † [Guo et al. (2019)]	73.1	64.2	69.0
Model with Pretrained Encoder			
TRE † [Alt et al. (2019)]	70.1	65.0	67.4
SpanBERT † [Joshi et al. (2019)]	70.8	**70.9**	70.8
Our Model			
Bi-DGA (ALBERT-base)	71.5	68.4	69.9
Bi-DGA (ALBERT-xlarge)	73.4	69.9	**71.6**

Table 1: Micro-averaged precision(P),recall(R), and F_1 score on TACRED dataset.† marks results reported in the original papers; ‡ marks results reported in [Zhang et al. (2017)] and [Zhang et al. (2018)].

as for SemEval2010. Batch size for training is 32. To avoid overfitting, we add a dropout before each Bi-DGA layer and the FFNN of classification layer with a rate of 0.1.

4.3. Results on TACRED Dataset

Tabel 1 shows the results of baseline as well as our proposed models on TACRED dataset. It is observerd that our proposed Bi-DGA model outperforms all baseline models by at least 0.8 F_1 and achieves a new state-of-the-art.

We can see from the table 1 that most models reach a high precision but a lower recall. One of the reasons is the unbalanced amount of negative training samples. The data skew leads to a biased prediction results. An effective way to solve this problem is to introduce external knowledge. The external knowledge can be seen as data augmentation which is not related to the data distribution of training set. The effect of external knowledge can be proven by the higher recall results of Knwl-attn model with knowledge from external lexical resources, the SpanBERT model as well as our Bi-DGA model with pretrained ALBERT.

Another important reason is information loss. Previous dependency based model only use the dependency path between entities on LCA tree which may lose important information. Thus, the sequence model and the C-GCN with a path-centric pruning strategy reach a higher recall than other dependency based models. But the C-GCN still suffer information loss from path-centric pruning as we demonstrate in section 1 . The C-AGGCN model improved this problem by a graph attention model based on fully dependency tree and reached higher recall and f1 score. But the sparse adjacency matrix is still a problem for C-AGGCN which results in a big gap between precision and recall.

Model	Macro-F_1
Sequence Model	
PA-LSTM‡ [Zhang et al. (2017)]	82.7
Knwl-attn† [Li et al. (2019)]	84.3
Dependency Based Model	
SDP-LSTM† [Xu et al. (2015)]	83.7
SPTree† [Miwa and Bansal (2016)]	84.4
C-GCN† [Zhang et al. (2018)]	84.8
C-AGGCN† [Guo et al. (2019)]	85.7
Pretrained Models	
TRE† [Alt et al. (2019)]	87.1
R-BERT† [Wu and He (2019)]	89.2
Our Model	
Bi-DGA - Label-to-Context Attention	89.5
Bi-DGA	**89.6**

Table 2: Macro-average F_1 score on SemEval2010-Task8 dataset. † marks results reported in the original papers; ‡ marks results reported in [Zhang et al. (2018)]. ALBERT-base is used as encoder for our model .

For our model, we encode the sentence by attention mechanism on the complete dependency tree with our information flow strategy and use a hop embedding to indicate the hop distance to dependency path. In this way, our model is able to capture long-range syntactic information between entities with less information loss.

4.4. Results on SemEval 2010 Dataset

We use SemEval 2010-Task8 dataset to evaluate the generalization ability of our proposed model. The results are shown in table 2. On the account of strong correlations between entities of interest and relation classes reported in previous work [Nguyen and Grishman (2015); Li et al. (2019)] , we experiment and report results without using the "entity mask" strategy which will degrades the performance. Different from TACRED dataset, the relation labels of SemEval are not specific relationships in the real world, but abstract semantic relationships. Thus, the result of Knwl-attn model using the external knowledge of labels is not so significant. For pretrained models, which provide rich semantic information, are particularly suitable for this dataset. Though the TRE and R-BERT have reached a very high F_1 score, we still improve the result by 0.3% with Our Bi-DGA, which prove the effectiveness of Bi-DGA on other dataset. By incorporating label-to-context attention, the result further improve 0.1% .

4.5. Ablation Study

To study the contibution of each components in our model, we perform ablation experiments on TACRED dataset. The results are shown in table 3. It is observed that: (1) The Bi-DGA model with label-to-context attention reach highest F_1 and recall and more

Model	P	R	F1
Bi-DGA (ALBERT-base)	71.5	68.4	69.9
- Label-to-Context Attention	72.3	66.7	69.4
- Direction	72.5	64.2	68.1
- Information Flow	**74.3**	62.3	67.8
- Dependency Tree	71.0	65.9	68.4
- Hop Embedding +Pruning	72.8	65.2	68.9
+ ALBERT-large	71.7	69.1	70.4
+ ALBERT-xlarge	73.4	**69.9**	**71.6**

Table 3: Ablation study on Bi-DGA model. Results are the median F_1 scores of 5 independent runs on test set of TACRED.

balanced P score and R score, which proves the label related context is indeed helpful for identifying correct positive relationships. (2) We remove the direction information by using an attention on all ancestors and all descendant nodes for each node without separating them. As we expect, and the precision and recall are both worsened. The information flow from different directions plays different roles, a mixture use of them will degrade the performance. (3) We further remove the information flow to do an attention only with adjacent nodes (only parent and children), which is more like a graph attention model with a sparse adjacency matrix, and reaches a worse result. In such a way, multiple layers stacking is required to interact with more nodes which may lead to information loss and confusion during layers stacking. (4) We even replace the Bi-DGA layers with self attention to remove the dependency information. Instead of hop embedding, we add the embedding of relative position to subject and object into the input representation, which are commonly used in previous works [Nguyen and Grishman (2015); Zhang et al. (2017)]. Without the dependency structure to guide the attention, the F_1 score is 1.0 lower than Bi-DGA, proving the importance of dependency information. Compared to the result of (2) and (3), we find that using the dependency information improperly gives the opposite effect. (5) We remove the hop embedding and use a K=1 path-centric pruning strategy same as [Zhang et al. (2018)], then the result decreases 1.0% . Through the experiment result , we can see that the path-centric pruning is still a aggressive strategy which may loss some helpful context information and lead to a decrease in recall and F_1 score. (6) To show the power of pretrained encoders with different size, we substitute the ALBERT-base with ALBERT-large and ALBERT-xlarge. We can observe that the result are significantly improved by larger pretrained encoders and reach a new state of art result with ALBERT-xlarge.

5. Conclusion

In this paper, we propose the bidirectional dependency guided attention for relation classification which is fully attention-based and taking the character of dependency tree as the starting point. Incorporating the label-to-context attention and ALBERT, the proposed model achieves state-of-the-art result on TACRED dataset and a significant result on SemEval2010-Taks8 dataset, showing superiority of our model to previous dependency

based models. In the future work, we will incorporate the conceptual knowledge of labels into label embedding to further enhance the effectiveness of label-to-context attention. Since we offer a way to encode tree structure information with attention mechanism, we will try to apply this idea to the more complicated transformer structure and to other NLP tasks such as event detection.

Acknowledgments

This research work was partially supported by CAS Key Lab of Network Data Science and Technology, ICT under Grant No.CASNDST202005, Beijing Academy of Artificial Intelligence under Grants No. BAAI2019ZD0306, and funded by National Natural Science Foundation of China under Grants No.61802029, 61722211, 61872338, and 61902381.

References

Christoph Alt, Marc Hbner, and Leonhard Hennig. Improving relation extraction by pre-trained language representations. In *Proceedings of AKBC 2019. Automated Knowledge Base Construction*, pages 1–18, 2019.

Rc Bunescu and Rj Mooney. A shortest path dependency kernel for relation extraction. In *Conference on Human Language Technology & Empirical Methods in Natural Language Processing*, 2005.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186, 2019.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 626–634, 2015.

Zhijiang Guo, Yan Zhang, and Wei Lu. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251, 2019.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, page 9499, 2009.

Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *CoRR*, abs/1606.08415, 2016.

Mandar Joshi, Danqi Chen, Y. Liu, Daniel S. Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2019.

N. KAMBHATLA. Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction. In *Meeting of Association of Computational Linguistics*, 2004.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. End-to-end neural coreference resolution. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 188–197, 2017.

Pengfei Li, Kezhi Mao, Xuefeng Yang, and Qi Li. Improving relation extraction with knowledge-attention. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 229–239, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *Computer Science*, 2013.

Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. *ArXiv*, abs/1601.00770, 2016.

Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015.

Zhou Peng, Shi Wei, Jun Tian, Zhenyu Qi, and Xu Bo. Attention-based bidirectional long short-term memory networks for relation classification. In *Meeting of the Association for Computational Linguistics*, 2016.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

Barbara Plank and Alessandro Moschitti. Embedding semantic similarity in tree kernels for domain adaptation of relation extraction. In *Meeting of the Association for Computational Linguistics*, 2015.

Alec Radford, Karthik Narasimhan, Time Salimans, and Ilya Sutskever. Improving language understanding with unsupervised learning. *Technical report, OpenAI*, 2018.

Adam Santoro, David Raposo, David G Barrett, Mateusz Malinowski, Razvan Pascanu, Peter Battaglia, and Timothy Lillicrap. A simple neural network module for relational reasoning. In *Advances in Neural Information Processing Systems 30*, pages 4967–4976. 2017.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pages 1556–1566, 2015.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.

Ngoc Thang Vu, Heike Adel, Pankaj Gupta, and Hinrich Schütze. Combining recurrent and convolutional neural networks for relation classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 534–539, 2016.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, 2016.

Shanchan Wu and Yifan He. Enriching pre-trained language model with entity information for relation classification. *CoRR*, 2019. URL http://arxiv.org/abs/1905.08284.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, 2015.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of Machine Learning Research*, pages 1083–1106, 2003.

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

Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *Computer Science*, 2015.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Conference on Empirical Methods in Natural Language Processing*, 2017.

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