Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.047811

ARTICLE

Graph Convolutional Networks Embedding Textual Structure Information for Relation Extraction

Chuyuan Wei*, **Jinzhe Li, Zhiyuan Wang, Shanshan Wan and Maozu Guo**

School of Electrical and Information Engineering, Beijing University of Civil Engineering and Architecture, Beijing, 102616, China

*Corresponding Author: Chuyuan Wei. Email: weichyuan@bucea.edu.cn

Received: 18 November 2023 Accepted: 19 February 2024 Published: 15 May 2024

ABSTRACT

Deep neural network-based relational extraction research has made significant progress in recent years, and it provides data support for many natural language processing downstream tasks such as building knowledge graph, sentiment analysis and question-answering systems. However, previous studies ignored much unused structural information in sentences that could enhance the performance of the relation extraction task. Moreover, most existing dependency-based models utilize self-attention to distinguish the importance of context, which hardly deals with multiple-structure information. To efficiently leverage multiple structure information, this paper proposes a dynamic structure attention mechanism model based on **textual structure information**, which deeply integrates word embedding, named entity recognition labels, part of speech, dependency tree and dependency type into a graph convolutional network. Specifically, our model extracts text features of different structures from the input sentence. **Textual Structure information Graph Convolutional Networks** employs the dynamic structure attention mechanism to learn multi-structure attention, effectively distinguishing important contextual features in various structural information. In addition, multi-structure weights are carefully designed as a merging mechanism in the different structure attention to dynamically adjust the final attention. This paper combines these features and trains a graph convolutional network for relation extraction. We experiment on supervised relation extraction datasets including SemEval 2010 Task 8, TACRED, TACREV, and Re-TACED, the result significantly outperforms the previous.

KEYWORDS

Relation extraction; graph convolutional neural networks; dependency tree; dynamic structure attention

1 Introduction

Relation Extraction (RE) aims to identify and extract the relation between two given entities in the input sentence. This task is vital in information extraction and has significant implications for various downstream natural language processing (NLP) applications, including sentiment analysis [1,2], question-answering systems [3] and text summarization [4]. As a critical and challenging task, how to improve the performance of RE has attracted considerable attention from researchers.

It is very important to fully exploit the different types of features in text to enhance the performance of the RE task [5–7]. To leverage rich feature information in the word sequences, many

RE models [8–13] have been proposed for extracting relations between entities. These models include recurrent neural network (RNN)-based approaches, long short-term memory (LSTM)-based models and transformer-based architecture methods. However, such models struggle to capture long-distance connections between words when modeling the linear sequence of text. Many studies utilize additional features and knowledge to deal with this problem. In all the options, dependency parses have been widely used and proven to be effective [14–17]. Dependency trees can provide long-distance word-word relations, which are essential supplementary structures for existing RE models. To effectively utilize dependency trees, most methods [15,16,18–20] employ graph convolutional networks (GCN) to model dependencies and extract relations between entities. Nevertheless, excessive reliance on dependency information could introduce confusion into RE [21–26]. Recently, Zhang et al. [15] combined a pruning strategy with GCN to model the dependency structure and perform RE. Tian et al. [20] proposed a new model that distinguishes important contextual information by dependency attention. These methods focus on utilizing the graph structure information within word sequences but do not leverage other important text inner features, such as part-of-speech (POS) labels and named entity recognition (NER) labels. This omission may impact the performance of the RE model.

Despite their effectiveness, existing methods have the following drawbacks:

1) Most previous studies [27–29] could not simultaneously utilize sequence-structure information and graph-structure information in the input text to extract the relation between entities. Some types of introduced sequence information in the model may help mitigate the effects of dependency noise. Such as the NER tags can provide entity features and build constrained relation between words, the POS tags can determine the function and feature of words, and the dependency trees can provide long-distance distances of words.

2) The attention mechanism in traditional research make it difficult to learn important information from multi-graph structures. Besides, pruning dependency tree strategies may introduce new noise to the dependency tree. These dependency trees are automatically extracted by the NLP toolkits. It is difficult to distinguish the noise by directly using dependency trees for modeling. Previous studies [7,15] have consistently required pruning strategies before utilizing dependency information for modeling. While some studies [20] employ self-attention mechanisms to distinguish dependency tree noise, they often focus on specific types of information which makes it challenging to discern noise in various dimensions.

To alleviate the impact of dependency tree noise on RE and effectively leverage textual inner features, we propose **Textual Structure information Graph Convolutional Networks** (**TS-GCN**). The model employs dynamic structure attention to learn the contextual feature weight from multiple types of information, filling the gap left by previous methods that did not simultaneously leverage both sequence information (such as POS type and NER type) and graph information (such as dependency trees and dependency type). We collectively refer to sequence information and graph information as '**Textual Structure Information**'. In addition, when there is noise in some structural information, the dynamic structural attention mechanism alleviates interference by adjusting the contextual attention weights for different structural information. Specifically, we first utilize the Standard CoreNLP Toolkits (SCT) to extract textual structure information from the input, then build various graphs based on the dependency tree to represent different textual structure information. Next, TS-GCN dynamically calculates the weights between words connected by dependency relation based on multiple graphs of text structure information, and finally utilizes dynamic weights to predict relations between entities. Besides, the TS-GCN dynamic distributes the weights among different graph structures based on information features, a crucial aspect often overlooked in previous studies, especially those employing attention mechanisms [18,20]. Experimental results on four English benchmark datasets—TACRED, TACREV, Re-TACRED, and SemEval 2010 Task 8—demonstrate the effectiveness of our RE approach using TS-GCN equipped with a dynamic structure attention mechanism. State-of-the-art performance is observed across all datasets.

The contribution of this paper can be summarized as follows:

1) A TS-GCN model based on textual structure information. This model can effectively model both sequential and graphical information within a sentence, realizing the extraction of entity relations.

2) We propose a dynamic structure attention mechanism aimed at mitigating the impact of dependency tree noise on relation extraction. This mechanism independently assigns weights to the feature connections within various text structure graphs. It then dynamically adjusts the contextual attention based on these individual connection weights, thereby mitigating the impact of the noise in structure (such as dependency tree noise, etc.) on relation extraction.

3) A relation modeling method is designed, which is based on multiple sources of structure information. By integrating sequence structure into the graph convolutional network, we create a multi-layered graph structure within the sentence, leading to a significant improvement in model performance.

2 Related Work

Early RE methods [30–33] typically relied on rule-based techniques or statistical mechanisms. These approaches heavily depended on the high-quality design of manually crafted features, and the effectiveness of the models was significantly influenced by the quality of these handcrafted features.

With the development of deep learning technology, neural network methods [34–38] excel in extracting semantic features embedding in text and have found widespread applications in RE tasks. Current RE models can be broadly categorized into two main types: Sequence-based and graph-based.

Sequence-based models [13,34], including CNNs, RNNs, and Transformers, employ neural networks to encode contextual information and capture latent features from word sequences. DNN [5] is recognized as one of the pioneering models that first introduced the use of CNNs for relation extraction, employing a convolutional method to acquire sentence features. Att-BLSTM [11] employed Bidirectional Long Short-Term Memory Networks (Bi-LSTM) to extract crucial semantic features from a sentence. It utilized an attention mechanism to capture associations between entities while taking the text context into account. This approach significantly enhanced the performance of relation extraction. SpanBERT [34] was a pre-training method specialized in predicting text spans. It achieved relation extraction by masking contiguous random spans within a given text and subsequently training the model based on representations of these span boundaries. This unique approach equipped SpanBERT with the ability to capture intricate contextual information within the text. Zhou et al. [13] introduced an innovative baseline approach for relation extraction, which integrates an entity representation technique. This technique was designed to effectively tackle the challenges associated with entity representation and ameliorate the influence of noisy or ambiguously defined labels. However, this modeling method faces challenges in effectively leveraging various knowledge sources, particularly the dependency tree and syntactic information.

Graph-based models, different from sequence-based models, leverage graph structure from dependency parsing information to capture long-distance contextual features. Currently, utilizing dependency trees for RE has become a mainstream trend. However, in most studies, dependency trees are automatically generated by toolkits, which may introduce some noise. Therefore, it is crucial to mitigate the impact of noise on RE. C-GCN [15] was the first to apply a graph convolutional network to relation extraction. It enabled effective aggregation of features from dependency structures, and the implementation of a novel path-centric pruning strategy designed to eliminate superfluous dependency information. C-GCN-MG [19] addressed cross-sentence n-ary relation extraction. It utilized a contextualized graph convolutional network spanning multiple dependent sub-graphs, and a method for building graphs around entities based on the dependency tree. A-GCN [20] leveraged dependency-type information and self-attention mechanisms to reduce the reliance on pruning strategies. RE-DMP [31] introduced multiple order dependency connections and types into the pre-training model to obtain an encoder equipped with dependency information. Zhang et al. [28] proposed a dual attention graph convolutional network (DAGCN) with a parallel structure. This network can establish multi-turn interactions between contextual and dependency information, simulating the multi-turn looking-back actions observed in human comprehension. Wu et al. [29] designed an engineering-oriented RE model based on Multilayer Perceptron (MLP) and Graph Neural Networks (GNN). This model replaces the information aggregation process in GCN s with MLP and achieves improved RE performance.

With the recent advancements in large language models (LLMs) in NLP, recent studies have often employed prompt learning or in-context learning (ICL) for RE tasks. However, most studies [39–43] indicate that most ICL models perform less effectively in relation extraction tasks, especially when the relation label space is extensive or the input sentence structure is complex, compared to traditional pre-train fine-tuning models. The performance of ICL in RE is influenced by various factors, including computational costs [40,41], prompt templates [42], LLM parameters [39], and constraints on input sequence length. These factors contribute to significant differences in the performance of the models. Yang et al. [43] observed when relation extraction task datasets already comprise rich and well-annotated data, with very few out-of-distribution examples in the test set, pre-train fine-tuned models consistently outperform ICL approaches. Longpre et al. [44] observed that the current upper limit of the capabilities of pre-train fine-tuning models has not been reached.

Although the graph-based studies mentioned above have made significant progress in the field of RE, they still have some shortcomings. On the one hand, some of the models [20,38] solely utilize the graph structure feature from the input for modeling, they fall short in comprehensively leveraging sequence-structure features. On the other hand, to mitigate the impact of the dependence noise on relation extraction, some models [18–20] utilize a self-attention mechanism based on word features and dependency types for extracting relations between entities, while other models [19,21] incorporate manually designed complex pruning methods to alleviate the impact of dependency tree noise. However, these methods face challenges in handling input information with multiple structural features and a large amount of noise.

Different from the existing RE models, our model has a dynamic structure attention mechanism to capture the important features from diverse structure information, thus alleviating the influence of dependency tree noise on the RE task. Additionally, our model deeply integrates POS types, NER types, dependency trees, and dependency types in RE tasks. In summary, our model is a textual structure model that effectively integrates various types of features and dynamically adjusts attention weights in textual structure information.

3 Proposed Methodology

3.1 Task Definition

A conventional method for relation extraction involves approaching it as a classification task. We propose TS-GCN, which leverages textual structure information to enhance the sparsity features of dependency matrices. This augmentation improves the ability of TS-GCN to distinguish dependency tree noise and enhances the performance of TS-GCN in relation extraction. In this study, we aim to mitigate the impact of dependency tree noise on the RE task. To achieve this, we propose a graph convolutional neural relation extraction model. This model is based on a dynamic structure attention mechanism, which operates within the framework of graph convolutional networks. Fig. 1 is the overall architecture of TS-GCN.

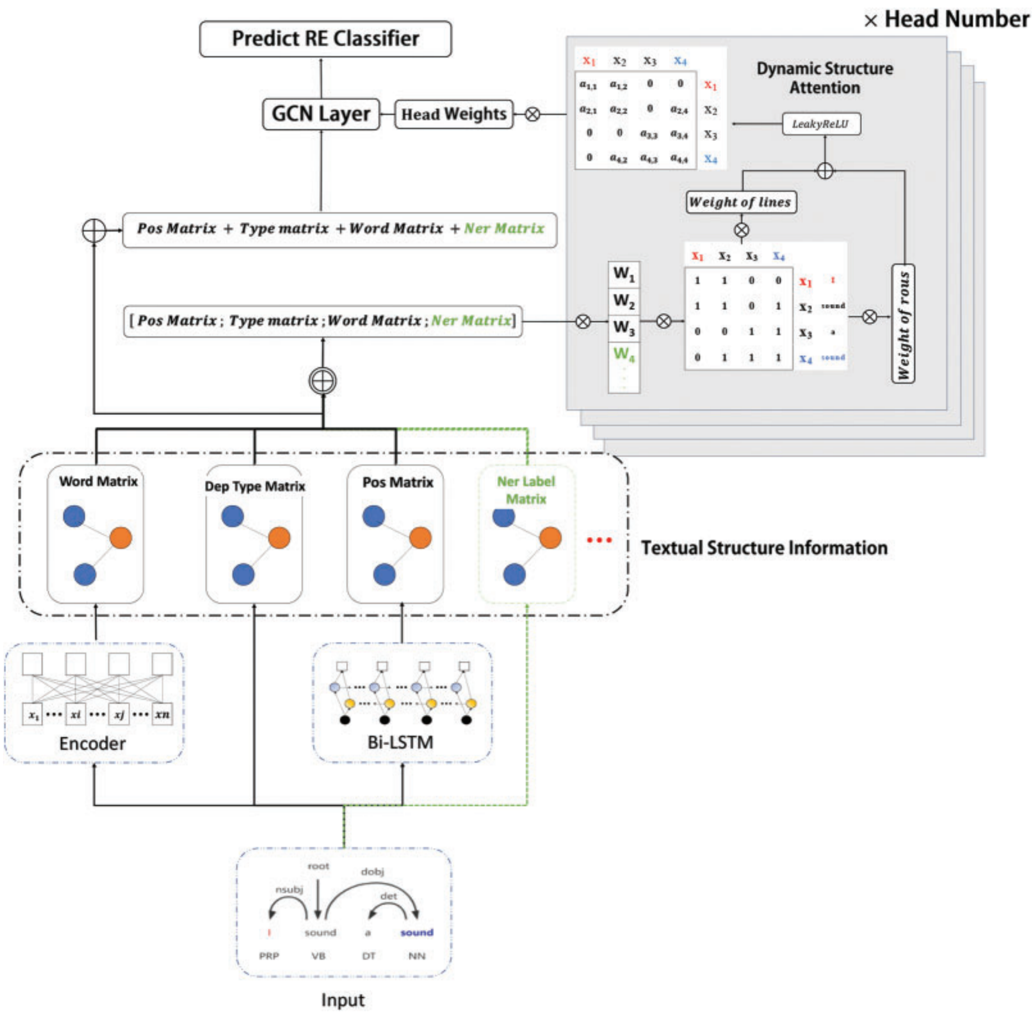


Figure 1: The overall architecture of our model TS-GCN for RE illustrated with an example input sentence (the two entities "I" and "sound" are highlighted in blue and red colors. Green color is coalesced other structure information into our model, example NER label)

Specifically, given an unstructured input sentence $\tau = x_1 \cdots x_n$ with n words, and two entity words (e_1, e_2) in a sentence. Then, utilize an off-the-shelf toolkit to obtain various textual structure information Z in τ. The prediction relation formula of τ between e_1 and e_2 in each sentence by

$$\tau = \operatorname{argmax}_{r \in R} P\left(r | TS - GCN\left(\tau, Z\right)\right) \tag{1}$$

where R is the relation set, Z is the textual structure information set, including POS, dependency types and NER labels. τ and A is the input of TS-GCN. The following sections begin by elaborating on the main components of our proposed TS-GCN and conclude by illustrating the process of applying TS-GCN to the classification paradigm for relation extraction.

3.2 Textual Structure Information Encoder

To enhance the reliability of dependency information, we combine word embedding, POS, dependency types and NER labels into a dependency matrix. As shown in Fig. 1, the data shown can be mined by toolkits from the given input sentence τ with n words (our datasets have given that information). After that we select word information τ, POS $\mathrm{p} = \{\mathrm{pos}_1, \cdots, \mathrm{pos}_n\}$ represents the corresponding POS features of the words and dependency type matrix $\mathrm{T_m} = (t_{i,j})_{n\times n}$ where $t_{i,j}$ is type class if two words $\mathrm{x_i}$ and $\mathrm{x_j}$ have dependency connection and otherwise $t_{i,j} = 0$. Due to POS information is inherently word dependent and exhibits a sequential contextual structure. The POS graph structure is not proficient at capturing these sequential features. In contrast to previous research, we employ Bi-LSTM to acquire contextual POS information, enabling us to learn contextual structure effectively.

$$\mathrm{pos} = \mathrm{Linear}\,(BiLSTM(p)) \tag{2}$$

where $\mathrm{pos} \in R^{n\times d}$ and d is the encoder's hidden dimension. Finally, we utilize the pos and x feature sequence to build respective matrices of learning contextual information $\mathrm{P_m} = (pos_{i,j})_{n\times n}$ and $\mathrm{X_m} = (x_{i,j})_{n\times n}$ where $p_{i,j} = \mathrm{pos}_i \times \mathrm{pos}_j$, $x_{i,j} = x_i \times x_j$. The matrices P_m, $\mathrm{X_m}$, and $\mathrm{T_m}$ are all $\mathrm{n} * \mathrm{n}$ dimension matrices.

3.3 TS-GCN

TS-GCN employs a novel approach to model word connections, distinct from the classic GCN-based model that assigns weights of either 0 or 1. We propose a dynamic structure attention mechanism to learn the node weights from different textual graphs. It allows the model to attend to diverse information across distinct structures simultaneously. This method can avoid interference from structure noise in the model. The structure attention mechanism can learn the bidirectional weights of dependency paths by considering the differences in text structure information among nodes.

First, we concatenate multiple input matrices P_m, $\mathrm{X_m}$, and $\mathrm{T_m}$ into a complete matrix Ts with multi-structure information and Ts is a $\mathrm{n} * \mathrm{n} * 3$ dimension matrix. Then, leverage the dot product of Ts and standard dependency connections matrix A to retain node information with dependency connections in the matrix Ts.

$$Ts = (\mathrm{P_m} \oplus \mathrm{X_m} \oplus \mathrm{X_m}) \times \mathrm{A} \tag{3}$$

where $\oplus$ denotes the vector concatenation operation, $Ts = (ts_{i,j})_{n\times n}$, A is the standard dependency matrix (if node exists dependency connection is 1, else 0).

The dynamic structure attention dynamically computes attention weights for Ts based on the combination of structure information. Next, set the weight w_h to dynamically filter the feature information in Ts and selectively enhance the representative information by

$$h_{i,j} = ts_{i,j} \times w_h \tag{4}$$

where $ts_{i,j} \in R^{3d}$, $w_h \in R^{3d} \rightarrow R^{3d}$. w_h is a learnable weight $3\mathrm{d} * 3\mathrm{d}$ dimension matrix. $\times$ is matrix multiplication. Different from previous methods, our structural attention is directional, i.e., $ts_{i,j} \neq ts_{j,i}$. This implies that during dynamic structure attention acquisition of contextual semantic information weight, distinct weights are assigned based on the direction of the dependency path by

$$\overrightarrow{att}^{head_t} = \mathrm{H} \cdot \overrightarrow{a}^{head_t} \tag{5}$$

where $\mathrm{H} = \left(h_{i,j}\right)_{n\times n}$, $\overrightarrow{a} \in R^{6d}$ and $head_t$ is the number of attention head. $\vec{a}$ is the forward dependency path attention weight from i to j. Afterwards, we aggregate forward $\overrightarrow{att}$ and reverse $\overleftarrow{att}$ to obtain the one-head attention matrix $a_{i,j}^{head_t}$ and compute the output of each one-head attention by

$$a_{i,j}^{head_t} = \mathrm{LeakyReL}\left(\overrightarrow{att}^{head_t} + \overleftarrow{att}^{head_t}\right) \tag{6}$$

$$attention_{i,j}^{head_t} = \mathrm{softmax}\left(a_{i,j}^{head_t}\right) = \frac{exp\left(a_{i,j}^{head_t}\right)}{\sum_{j\in n} exp\left(a_{i,j}^{head_t}\right)} \tag{7}$$

Finally, we utilize the head weight w_a^t to normalize dynamic structure attention by

$$atten = \sum\nolimits_{t\in head} attention^t \times w_a^t \tag{8}$$

where $w_a^t \in R^{head} \rightarrow R$ and w_a^t is a head∗1 dimension matrix. We apply the attention $atten_{i,j}$ to the commonly associated connection between Th_i and Th_j and obtain the output representation of o_i by

$$Th_{i,j} = \left(P_{mi,j} + X_{mi,j} + T_{mi,j}\right) \cdot \mathrm{W} \tag{9}$$

$$o_i = \mathrm{ReLU}\left(\sum\nolimits_{j\in n}\left(atten_{i,j} \cdot Th_{i,j}\right) + b\right) \tag{10}$$

where $\mathrm{W} \in R^d \rightarrow R^d$, W is a n ∗ n dimension paranoid matrix. Compared with traditional GCN, TS-GCN uses dynamic structure attention to dynamic weights to distinguish the importance of different structure content. This method helps the model more fully understand and leverage complex textual structure information. Furthermore, our approach allows for the incorporation of additional features, such as NER labels, etc., into the textual structure information. This approach enhances the extensibility and convenience of our model, enabling the exploration of additional textual structure information.

3.4 Relation Extraction with TS-GCN

Before employing TS-GCN for RE, we firstly employ BERT [45] to encode the input x into hidden embedding, with $t_i^{(0)}$ representing the hidden embeddings for x_i. We next apply our proposed TS-GCN with N layers to obtain the corresponding output o_i^n based on the input $t_i^{(0)}$. Then, we employ the max pooling mechanism to obtain the output hidden embeddings t_e $(e = subj, obj)$ for the entity words by

$$t_e = \mathrm{MaxPooling}\left(\left\{t_i^{(N)} | i = subj, obj\right\}\right) \tag{11}$$

Afterward, we utilize matrix multiplication on the concatenated embeddings of the two entities using the trainable matrix W_p and apply the ReLU activation function to obtain the output embedding by

$$P(r) = \frac{exp\left(W_r z + b_r\right)}{\sum_{r'\in|R|} exp\left(W_{r'} z + b_{r'}\right)} \tag{12}$$

where $W_r \in R^d$, $W_{r'} \in R^d$, $b_r \in R$, $b_{r'} \in R$ and $|R|$ is the type of relation. W_r and $W_{r'}$ are d dimension paranoid matrixes. b_r and $b_{r'}$ are paranoid weights.

4 Experiments and Analyses

4.1 Preliminary

Datasets. We use four English datasets in the experiments including SemEval 2010 Task 8 (SemEval) [46] and three versions of TACRED: The original TACRED [12], TACREV [47], and Re-TACRED [48]. Due to the presence of approximately 6.62% noisily labeled instances in the TACREV dataset, Alt et al. [47] relabeled it using the TACRED development and test set, and Stoica et al. [48] relabeled the whole dataset by further refining the label definitions on TACRED. For SemEval, we use its official train/test split. We provide the statistics of the datasets in Table 1.

Table 1: The statistics of datasets

Dataset	Train	Dev	Test	Class
SemEval	8000	–	2717	10
TACRED	68124	22631	15509	42
TACREV	68124	22631	15509	42
Re-TACRED	58465	19584	13418	40

4.2 Results and Discussion

Model configurations. We follow the study of Soares et al. [49] to insert four special tokens, which are "e1", "/e1", "e2", and "/e2" into the input sentence to mark the boundary of the two entities. This strategy allows the encoder to distinguish the position of entities during encoding and improves model performance. For the encoder, we utilize the uncased versions of BERT-base and BERT-large [45] from HuggingFace, while following the default settings. Our model is optimized with Adam [50] using the learning rate of 7E-6 on BERT-base and BERT-large, setting four-head dynamic structure attention to obtain important representations. We evaluate all combinations of each model and use the one with the best performance (i.e., F1 scores) on the development set.

Evaluation. For SemEval, we follow previous studies and use the official evaluation to evaluate it. (The official evaluation script downloaded from https://huggingface.co/datasets/sem_eval_2010_task_8/blob/main/sem_eval_2010_task_8.py).

For three versions of TACRED, we use the mainstream evaluation formula, P, R and Micro-F1(F1).

$$P = \frac{1}{C}\sum_{i=1}^{C} P_i \times 100\% \tag{13}$$

$$R = \frac{1}{C}\sum_{i=1}^{C} R_i \times 100\% \tag{14}$$

$$F_1 = \frac{2 \times (P \times R)}{P + R} \tag{15}$$

where C is the relation class, P_i is the precision score results in class i and R is the recall score results in class i.

Baseline. We compare TS-GCN on Bert-Large and Bert-Base with the state-of-the-art sentence-level relation extraction model proposed by Tian et al. [20]. They utilized dependency types to acquire attention in dependency nodes, which represent the importance of the node in the information matrix.

We follow Tian et al. [20] given the best default settings to train their model on three versions of TACRED, since they only showed the best F1 points on SemEval.

Furthermore, TS-GCN demonstrates performance improvements on the TACRED, TACREV, and RETACRED datasets. We also conducted a comparison with the latest baseline model by Zhou et al. [13], which is based on a transformer architecture.

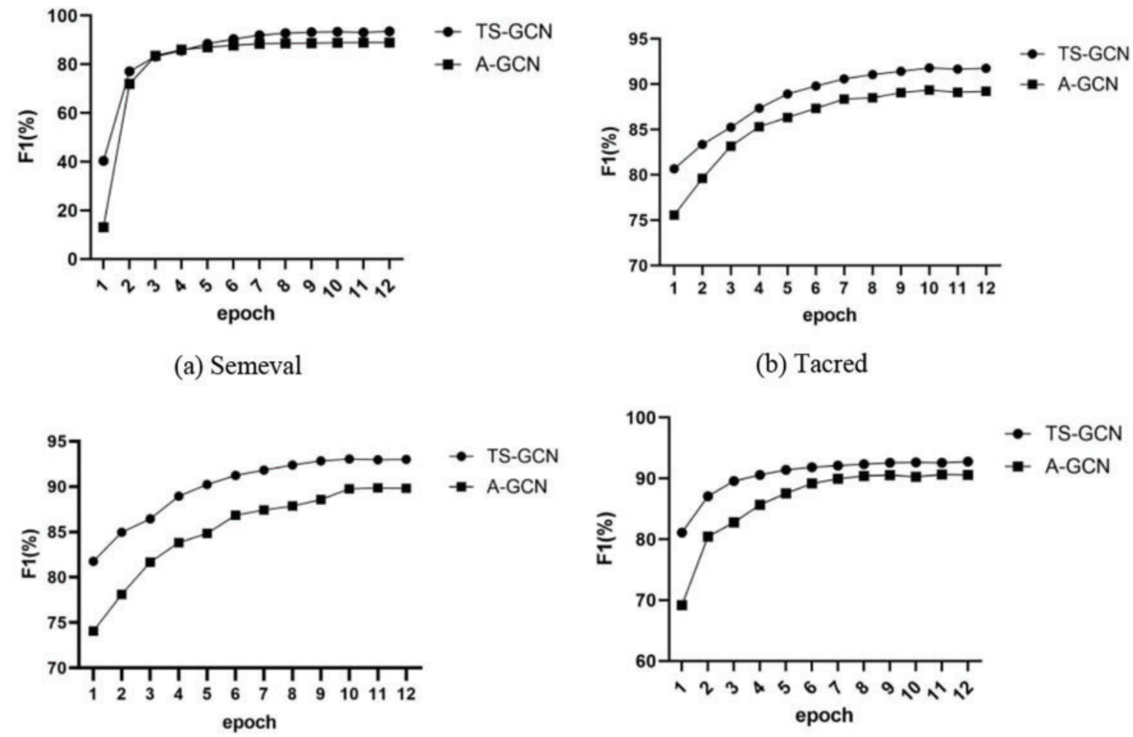
Table 2 shows the comparison of our TS-GCN approach to the baseline, which uses dependency-driven relation extraction and other studies. Our approach outperforms the baseline methods on the four datasets. Especially on the TACRED dataset, our approach achieves an F1 score of 87.73% and 88.28%, which is significantly higher than the baseline model of 86.76% and 87.64% by Tian et al. [20] and achieves a new SOTA compared to previous studies such as 72.9% by Zhou et al. [13], 70.8% by Joshi et al. [27], 66.3% by Zhang et al. [15]. This proves our method can bring consistent and considerable performance improvements to all the datasets. Besides, when utilizing BERT-BASE as the encoder, TS-GCN still achieves state-of-the-art (SOTA) performance on four datasets. One the one hand, this indicates that TS-GCN effectively learns the representations of textual structure information in the input text, which reduces the impact of noise in dependency trees on relation extraction. On the other hand, it demonstrates that the performance enhancement in TS-GCN does not result from the encoder replacement.

Table 2: Result of F1 scores (in %) between previous studies and our best models

Models	SemEval	TACRED			TACREV			Re-TACRED		
	F_1	P	R	F_1	P	R	F_1	P	R	F_1
BERTEM+MTB [49]	89.5	71.8	68.4	70.1	–	–	–	–	–	–
LST-AGCN [1]	86.0	69.6	68.0	68.8	–	–	–	–	–	–
C-GCN-MG [19]	85.9	67.1	65.1	66.1	–	–	–	–	–	–
PA-LSTM [12]	–	66.6	63.6	65.1	74.5	72.1	73.3[3]	82.1	76.8	79.4[2]
C-GCN [15]	84.8	87.2	82.5	84.8	84.9	84.7	84.8[3]	86.6	83.1	84.8[2]
SpanBERT [27]	–	71.2	70.3	70.8	80.5	75.6	78.0[1]	85.8	84.7	85.3[2]
RE-Improved ($BERT_{LARGE}$) [13]	–	–	–	72.9	–	–	81.3	–	–	89.7
DAGCN [28]		72.4	64.8	68.4	–	–	–	–	–	–
Wu et al. [29]	83.5	71.1	62.8	66.7	–	–	–	–	–	–
A-GCN ($BERT_{BASE}$) [20]	89.16	87.87[4]	83.74[4]	85.76[4]	88.75[4]	87.12[4]	87.94[4]	90.01[4]	86.51[4]	88.23[4]
A-GCN ($BERT_{LARGE}$) [20]	89.85	88.73[4]	84.63[4]	86.64[4]	89.40[4]	87.06[4]	88.22[4]	92.31[4]	86.71[4]	89.43[4]
Our Model										
TS-GCN ($BERT_{BASE}$)	89.86	90.41	85.20	87.73	92.41	89.09	90.72	92.93	87.86	90.33
TS-GCN ($BERT_{LARGE}$)	**91.61**	89.78	86.82	**88.28**	93.37	90.33	**91.81**	93.76	88.56	**91.09**

Note: [1] Marks re-implemented results from Alt et al. [47]. [2] Marks re-implemented results from Stoica et al. [48]. [3] Marks re-implemented results from Zhou et al. [13]. [4] Marks our re-implemented results.

In Fig. 2, we present the F1 score progression of TS-GCN with the increasing number of epochs. It shows that TS-GCN reaches convergence faster than the A-GCN baseline model. This is a significant advantage because it means that the model can be trained more efficiently, which saves both time and resources. Additionally, the figure shows that TS-GCN consistently achieves faster convergence results than A-GCN on all four datasets that we tested. This finding confirms that TS-GCN is a robust and effective model that can be applied to a wide range of NLP tasks with high accuracy.



(a) Semeval (b) Tacred

(c) Tacrev (d) Re-Tacred

Figure 2: The contrasted F1 scores for four datasets were obtained during training using the BERT-large encoder

Overall, all evaluation demonstrates that TS-GCN is a powerful and efficient model for RE. Its ability to reach convergence quickly and achieve a higher F1 score than the A-GCN baseline model makes it an excellent choice for RE.

4.3 Ablation Study

To further analyze TS-GCN, we conduct an ablation study on best model to study the effectiveness of each component on four datasets. Compared to the previous RE model that applies GCN, TS-GCN enhances the semantic exploration ability in two aspects: 1) using a Bi-directional long short-term memory (Bi-LSTM) to enrich the representation of POS representations and enhance the sensitivity to context, 2) introducing multi-head dynamic structure attention to weight different textual structure information for reducing the impact of dependency tree noise interference, To investigate the independent enhancement effects of each modules, we conduct an ablation study on our best model. The best model includes two layers of TS-GCN, 4 heads of dynamic structure attention, and utilizes dependency type and POS information.

Table 3 shows the experimental results of different modules, including the performance of the GCN baseline and the BERT-only baseline for reference. The results indicate the ablation of modules could result in worse results. Especially, the ablation of the multi-head dynamic structure attention module significantly impairs TS-GCN. This indicates that the guided learning of information is abandoned, making TS-GCN susceptible to dependency tree noise, making it difficult to learn correct features.

Table 3: The ablation study results (F1) of TS-GCN on whether use dynamic structure attention mechanism and POS Bi-LSTM. '✓' and '×' stand for that whether a module is used

	Bi-LSTM	G-ATT	SemEval	TACRED	TACREV	Re-TACRED
$BERT_{BASE}$	GCN		88.62	83.35	86.21	86.75
TS-GCN	✓	✓	89.86	87.73	90.10	90.33
	×	✓	89.21	86.51	89.55	89.87
	✓	×	88.03	85.79	88.13	88.36
	Only Bert		87.87	71.56	79.33	85.91
$BERT_{LARGE}$	GCN		89.13	84.95	86.68	87.02
TS-GCN	✓	✓	91.61	88.28	91.81	91.09
	×	✓	89.56	87.22	90.22	90.26
	✓	×	88.39	86.98	89.88	89.18
	Only Bert		89.02	72.95	81.31	86.72

Table 4 shows the experimental results of textual structure information with different feature combinations, which include dependency types (Dep), POS labels (POS), and NER labels (NER). The results indicate that an increase in the types of textual structure information leads to improved performances. Multiple types of textual structure data are important to TS-GCN, especially with some noise in input information. Without the introduction of POS and NER features, the F1 performance of TS-GCN using BERT-Base decreases by 0.88%, 0.93%, and 1.12%, while using BERT-Large, it decreases by 0.89%, 1.53%, and 0.55%. This illustrates that our method effectively mitigates the impact of noise in the dependency tree on context learning, leading to improved results in relation extraction.

Table 4: The ablation study results (F1) of TS-GCN on textual structure information module

	Dep	POS	NER	TACRED	TACREV	Re-TACRED
$BERT_{BASE}$	✓	✓	✓	88.24	91.71	91.04
TS-GCN	✓	✓	×	87.73	90.72	90.33
	✓	×	×	87.36	89.75	89.92
$BERT_{LARGE}$	✓	✓	✓	88.72	92.26	91.38
TS-GCN	✓	✓	×	88.28	91.81	91.09
	✓	×	×	87.83	90.73	90.83

4.4 Case Study

To investigate the effect of the number of heads in dynamic structure attention on TS-GCN, we conducted a case study using our TS-GCN models with different numbers of dynamic structure attention heads.

Table 5 shows the experimental results with different numbers of dynamic structure attention heads, including 1, 2, 4, 8, and 12. In this table, we observe that the 4-head dynamic structure attention obtains better performance compared to the 1-head and 2-head configurations. Furthermore, compared to the 8- head and 12-head configurations, the 4-head configuration requires fewer computing resources and achieves similar optimal performance. Therefore, using 4 heads can enhance the training efficiency of our model. Overall, we conclude that the optimal configuration for multi–head dynamic structure attention is 4.

Table 5: The case study results (F1) for multi-head dynamic structure attention (BERT-Large), S is train computing resources consumed by TS-GCN

Head Num	SemEval		TACRED		TACREV		Re-TACRED	
	F_1	S (GB)	F_1	S (GB)	F_1	S (GB)	F_1	S (GB)
1 head	89.37	2.4	86.10	3.6	89.05	2.9	89.08	3.2
2 head	90.16	5.1	87.09	7.7	90.33	6.6	90.09	6.9
4 head	91.61	9.1	88.28	15.9	91.81	14.5	91.09	15.3
8 head	91.36	14.0	88.36	21.4	91.73	19.8	90.40	20.0
12 head	91.46	20.6	87.94	30.0	90.82	28.3	91.11	28.8

To investigate the effect of the number of layers in TS-GCN on RE, we conducted a case study by training our model with different numbers of layers.

Table 6 shows the experimental results on the test datasets for TS-GCN with varying numbers of layers, 1, 2, 3, and 4. In this table, we can observe that the TS-GCN with 2 layers precedes other configurations. We consider this result to be due to the ease with which the weights of the multi-head dynamic structure attention can be influenced by the number of convolutional layers. When the number of layers is set to 1, it is difficult for TS-GCN to learn deep contextual features. On the other hand, when the number of layers exceeds 2, the multi-head dynamic structure attention weight of TS-GCN becomes averaged, which makes it less sensitive to noise in the input text. Overall, we conclude that the optimal configuration for TS-GCN is with 2 layers.

Table 6: The case study results (F1) on varying the number of layers in TS-GCN (BERT-Large), S is train computing resources consumed by TS-GCN

Layers Num	SemEval		TACRED		TACREV		Re-TACRED	
	F_1	Size (GB)	F_1	Size (GB)	F_1	Size (GB)	F_1	Size (GB)
1 Layers	88.67	4.7	87.16	7.5	90.00	6.1	88.76	7.4
2 Layers	91.61	9.1	88.28	15.9	91.81	14.5	91.09	15.3

(Continued)

Table 6 (continued)

Layers Num	SemEval		TACRED		TACREV		Re-TACRED	
	F_1	Size (GB)	F_1	Size (GB)	F_1	Size (GB)	F_1	Size (GB)
3 Layers	90.87	14.5	88.03	23.6	91.22	21.8	90.71	22.9
4 Layers	89.60	18.8	87.77	31.1	90.96	30.1	89.85	30.7

To investigate the resistance to dependency tree noise interference of TS-GCN, we conducted a case study by randomly masking some of the dependency nodes in the test sets.

Table 7 shows the test results of experiments conducted on the best-trained model having noisy testsets. These test sets had 5%, 10%, and 20% of their dependent connections randomly removed, respectively. The results indicate that when the noise proportion is less than or equal to 10%, there is no significant decrease in our model performance. We believe that textual structure information can enhance the capacity of our model for self-correction. Furthermore, the dynamic structure attention mechanism adapts the contextual attention weights based on distinct information characteristics, thereby mitigating the interference of dependency tree noise in the context of RE. Overall, we conclude that TS-GCN with strong resilience to noise features.

Table 7: The case study results (F1) on the noise resistance of TS-GCN (BERT-Large)

Mask Pct	SemEval	TACRED	TACREV	Re-TACRED
	F_1	F_1	F_1	F_1
0%	91.61	88.28	91.81	91.09
5%	90.49	88.04	90.97	89.24
10%	89.87	87.84	89.58	88.21
20%	88.91	85.47	84.52	86.76

5 Conclusion

In this paper, we propose a graph convolutional network embedding textual structure information for relation extraction. We transform the task into a multi-information graph structure problem by incorporating different sequence information into graph nodes and propose a TS-GCN model that utilizes a dynamic structure attention mechanism to learn the importance of contextual information on dependency tree paths. This attention-learning process is dynamic and will selectively highlight and express important path information according to the composition of structural information features. Furthermore, we assign different learning weights to all information graph structures to reduce the impact of noise generated during the generation of information graphs on relation extraction. Experiments are conducted on the popular TACRED dataset, TCREV dataset, Re-TACRED dataset and SemEval 2010 Task 8 dataset. The results demonstrate that TS-GCN surpasses the best existing GCN-based models on the four datasets. We demonstrate that TS-GCN is a multiple-structure attention method, which emphasizes the importance of textual structure information in concerning extraction. To validate our approach, we conduct ablation experiments on the proposed dynamic structure attention mechanism and additional textual structure information. The experimental results show that increasing the types of information can mitigate the impact of dependency noise on relation extraction. Dynamic structure attention can improve the ability of the model to effectively learn multiple structure information. However, the size of the TS-GCN model will increase significantly as the number of attention heads and the number of graph convolution layers increases, but the model performance gradually levels off. Although our model achieves satisfactory results in representing relation extraction with graph neural networks, there is still significant study room for future work. Specifically, we plan to propose a more generalizable model template that minimizes the training cost of the model when introducing new textual structure information. Additionally, a meaningful direction is to compress the existing TS-GCN model to reduce computational costs. The dependency tree matrix is often a sparse matrix with huge computational costs, which presents a challenging yet important problem. We also would like to explore LLM and combine textual structure information to learn contextual features and enhance the performance of relation extraction. In addition, relation extraction can be combined with technologies such as knowledge graphs to provide technical support for practical problems in many industrial fields. For example, it helps to construct an intelligent knowledge graph belonging to industrial parts or manufacturing processes and infers whether the part is a qualified part through external information such as the size of the shape. Such research can provide application directions for GCN-based relation extraction methods and promote the further development of relation extraction technology.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Study conception and design: Chuyuan Wei, Jinzhe Li; data collection: Zhiyuan Wang; analysis and interpretation of results: Jinzhe Li, Shanshan Wan; draft manuscript preparation: Chuyuan Wei, Maozu Guo. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The TACRED data used in the study are publicly available for purchase at https://catalog.ldc.upenn.edu/LDC2018T24. The SemEval data used in the study is publicly available in https://huggingface.co/datasets/sem_eval_2010_task_8.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] K. Sun, R. Zhang, S. Mensah, Y. Mao, and X. Liu, "Aspect-level sentiment analysis via convolution over dependency tree," in *Proc. EMNLP-IJCNLP*, Hong Kong, China, 2019, pp. 5679–5688.

[2] A. Khan, H. Zhang, N. Boudjellal, A. Ahmad, and M. Khan, "Improving sentiment analysis in election-based conversations on twitter with elecbert language model," *Comput. Mater. Contin.*, vol. 76, no. 3, pp. 3345–3361, 2023. doi: 10.32604/cmc.2023.041520.

[3] K. Xu, S. Reddy, Y. Feng, S. Huang, and D. Zhao, "Question answering on Freebase via relation extraction and textual evidence," in *Proc. ACL*, Berlin, Germany, 2016, pp. 2326–2336.

[4] L. Wang and C. Cardie, "Focused meeting summarization via unsupervised relation extraction," in *Proc. SIGDIAL*, Seoul, South Korea, 2012, pp. 304–313.

[5] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation classification via convolutional deep neural network," in *Proc. COLING*, Dublin, Ireland, 2014, pp. 2335–2344.

[6] D. Zhang and D. Wang, "Relation classification via recurrent neural network," arXiv preprint arXiv: 1508.01006, 2015.

[7] Y. Xu, L. Mou, G. Li, Y. Chen, H. Peng and Z. Jin, "Classifying relations via long short term memory networks along shortest dependency paths," in *Proc. EMNLP*, Lisbon, Portugal, 2015, pp. 1785–1794.

[8] C. dos Santos, B. Xiang, and B. Zhou, "Classifying relations by ranking with convolutional neural networks," in *Proc. ACL-IJCNLP*, Beijing, China, 2015, pp. 626–634.

[9] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional long short-term memory networks for relation classification," in *Proc. PACLIC*, Shanghai, China, 2015, pp. 73–78.

[10] L. Wang, Z. Cao, G. de Melo, and Z. Liu, "Relation classification via multi-level attention CNNs," in *Proc. ACL*, Berlin, Germany, 2016, pp. 1298–1307.

[11] P. Zhou *et al.*, "Attention-based bidirectional long short-term memory networks for relation classification," in *Proc. ACL*, Berlin, Germany, 2016, pp. 207–212.

[12] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proc. EMNLP*, Copenhagen, Denmark, 2017, pp. 35–45.

[13] W. Zhou and M. Chen, "An improved baseline for sentence-level relation extraction," in *Proc. AACL-IJCNLP*, 2022, pp. 161–168.

[14] M. Miwa and M. Bansal, "End-to-end relation extraction using LSTMs on sequences and tree structures," in *Proc. ACL*, Berlin, Germany, 2016, pp. 1105–1116.

[15] Y. Zhang, P. Qi, and C. D. Manning, "Graph convolution over pruned dependency trees improves relation extraction," in *Proc. EMNLP*, Brussels, Belgium, 2018, pp. 2205–2215.

[16] K. Sun, R. Zhang, Y. Mao, S. Mensah, and X. Liu, "Relation extraction with convolutional network over learnable syntax-transport graph," in *Proc. AAAI*, New York, USA, 2020, vol. 34, pp. 8928–8935.

[17] G. Chen, Y. Tian, Y. Song, and X. Wan, "Relation extraction with type-aware map memories of word dependencies," in *Proc. ACL-IJCNLP*, 2021, pp. 2501–2512.

[18] Z. Guo, Y. Zhang, and W. Lu, "Attention guided graph convolutional networks for relation extraction," in *Proc. ACL*, Florence, Italy, 2019, pp. 241–251.

[19] A. Mandya, D. Bollegala, and F. Coenen, "Graph convolution over multiple dependency sub-graphs for relation extraction," in *Proc. COLING*, Barcelona, Spain, 2020, pp. 6424–6435.

[20] Y. Tian, G. Chen, Y. Song, and X. Wan, "Dependency-driven relation extraction with attentive graph convolutional networks," in *Proc. ACL-IJCNLP* , 2021, pp. 4458–4471.

[21] B. Yu, M. Xue, Z. Zhang, T. Liu, Y. Wang, and B. Wang, "Learning to prune dependency trees with rethinking for neural relation extraction," in *Proc. COLING*, Barcelona, Spain, 2020, pp. 3842–3852.

[22] J. Zhang, K. Hao, X. S. Tang, X. Cai, Y. Xiao and T. Wang, "A multi-feature fusion model for Chinese relation extraction with entity sense," *Knowl.-Based Syst.*, vol. 206, no. 7, pp. 0950–7051, 2020. doi: 10.1016/j.knosys.2020.106348.

[23] A. Javeed, "A hybrid attention mechanism for multi-target entity relation extraction using graph neural networks," *Mach. Learn. Appl.*, vol. 11, pp. 2666–8270, 2023. doi: 10.1016/j.mlwa.2022.100444.

[24] Q. Wan, S. Du, Y. Liu, J. Fang, L. Wei and S. Liu, "A hybrid attention mechanism for multi-target entity relation extraction using graph neural networks," *Mach. Learn. Appl.*, vol. 11, pp. 2666–8270, 2023.

[25] J. Liao, Y. Du, J. Hu, H. Li, X. Li and X. Chen, "A contextual dependency-aware graph convolutional network for extracting entity relations," *Expert Syst. Appl.*, vol. 239, no. 10, pp. 0957–4174, 2024. doi: 10.1016/j.eswa.2023.122366.

[26] K. Du, B. Yang, S. Wang, Y. Chang, S. Li and G. Yi, "Relation extraction for manufacturing knowledge graphs based on feature fusion of attention mechanism and graph convolution network," *Knowl.-Based Syst.*, vol. 255, no. 2, pp. 0950–7051, 2022. doi: 10.1016/j.knosys.2022.109703.

[27] N. Wang, T. Chen, C. Ren, and H. Wang, "Document-level relation extraction with multi-layer heterogeneous graph attention network," *Eng. Appl. Artif. Intell.*, vol. 123, no. 4, pp. 106212, 2023. doi: 10.1016/j.engappai.2023.106212.

[28] D. Zhang, Z. Liu, W. Jia, F. Wu, H. Liu and J. Tan, "Dual attention graph convolutional network for relation extraction," *IEEE Trans. Knowl. Data Eng.*, vol. 36, no. 2, pp. 1588–2191, 2023. doi: 10.1109/TKDE.2023.3289879.

[29] T. Wu, X. You, X. Xian, X. Pu, S. Qiao and C. Wang, "Towards deep understanding of graph convolutional networks for relation extraction," *Data Knowl. Eng.*, vol. 149, pp. 102265, 2023.

[30] N. Kambhatla, "Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction," in *Proc. ACL Interact. Poster Demon. Sessions, Assoc. Comput. Linguist.*, Barcelona, Spain, 2004, pp. 178–181.

[31] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proc. ACL,* Ann Arbor, Michigan, USA, 2005, pp. 427–434.

[32] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *J. Mach. Learn. Res.*, vol. 3, pp. 1083–1106, 2003.

[33] C. Aone, L. Halverson, T. Hampton, and M. Ramos-Santacruz, "SRA: Description of the IE_2 system used for MUC-7," in *Proc. MUC-7*, Virginia, USA, Apr. 29–May 1, 1998.

[34] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer and O. Levy, "SpanBERT: Improving pre-training by representing and predicting spans," *Trans. Assoc. Comput. Linguist.*, vol. 8, pp. 64–77, 2020. doi: 10.1162/tacl_a_00300.

[35] K. Ding *et al.*, "A knowledge-enriched and span-based network for joint entity and relation extraction," *Comput. Mater. Contin.*, vol. 68, no. 1, pp. 377–389, 2021. doi: 10.32604/cmc.2021.016301.

[36] D. Zeng, Y. Xiao, J. Wang, Y. Dai, and A. Kumar Sangaiah, "Distant supervised relation extraction with cost-sensitive loss," *Comput. Mater. Contin.*, vol. 60, no. 3, pp. 1251–1261, 2019. doi: 10.32604/cmc.2019.06100.

[37] L. Yin, X. Meng, J. Li, and J. Sun, "Relation extraction for massive news texts," *Comput. Mater. Contin.*, vol. 60, no. 1, pp. 275–285, 2019. doi: 10.32604/cmc.2019.05556.

[38] Y. Tian, Y. Song, and F. Xia, "Improving relation extraction through syntax-induced pre-training with dependency masking," in *Find. Assoc. Comput. Linguist.: ACL 2022, Assoc. Comput. Linguist.*, Dublin, Ireland, 2022, pp. 1875–1886.

[39] Z. Wan *et al.*, "GPT-RE: In-context learning for relation extraction using large language models," arXiv preprint arXiv:2305.02105, 2023.

[40] X. Xu, Y. Zhu, X. Wang, and N. Zhang, "How to unleash the power of large language models for few-shot relation extraction?" in *Proc. SustaiNLP*, Toronto, Canada, 2023, pp. 190–200.

[41] Y. Ozyurt, S. Feuerriegel, and C. Zhang, "In-context few-shot relation extraction via pre-trained language models," arXiv preprint arXiv:2310.11085, 2023.

[42] C. Peng *et al.*, "Model tuning or prompt tuning? A study of large language models for clinical concept and relation extraction," arXiv preprint arXiv:2310.06239, 2023.

[43] J. Yang *et al.*, "Harnessing the power of LLMs in practice: A survey on ChatGPT and beyond," arXiv preprint arXiv:2304.13712, 2023.

[44] S. Longpre *et al.*, "The flan collection: Designing data and methods for effective instruction tuning," arXiv preprint arXiv:2301.13688, 2023.

[45] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, Minneapolis, Minnesota, USA, 2019, pp. 4171–4186.

[46] I. Hendrickx *et al.*, "SemEval-2010 Task 8: Multi-way classification of semantic relations between pairs of nominals," in *Proc. SemEval*, Uppsala, Sweden, 2010, pp. 33–38.

[47] C. Alt, A. Gabryszak, and L. Hennig, "TACRED revisited: A thorough evaluation of the TACRED relation extraction task," in *Proc. ACL*, 2020, pp. 1558–1569.

[48] G. Stoica, E. A. Platanios, and B. Poczos, "Re-TACRED: Addressing shortcomings of the TACRED dataset," in *Proc. AAAI*, vol. 35, no. 15, pp. 13843–13850, 2021. doi: 10.1609/aaai.v35i15.17631.

[49] L. B. Soares, N. FitzGerald, J. Ling, and T. Kwiatkowski, "Matching the blanks: Distributional similarity for relation learning," in *Proc. ACL*, Florence, Italy, 2019, pp. 2895–2905.

[50] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.