Att-FMI: A Fusing Multi-Information Model with Self-Attentive Strategy for Relation Extraction

Kailiang Wang, Xuefeng Fu*, Yanping Liu, WeiKun Chen, and Jun Chen

***Abstract*—Relation classification is an important task in information extraction that involves identifying potential semantic connections between two entities in a sentence. Most relation extraction models either use semantic or structure information as the relation representation for classification. Although some researchers have tried to fuse both types of information, they simply combine different information as relation representation, which ignores semantic and structure information contribute unequally to the relation representation at the instance level, leading to degradation of performance and generalization capability. To address this issue, we propose a fusing multi-information model with a self-attentive strategy (Att-FMI) for relation classification. Our approach utilizes the pre-trained BERT model and dependency syntactic parser to obtain semantic and structure representations from the sequence, respectively and efficiently combines the two with support the self-attention mechanism. Experimental results on four commonly used datasets, including TACRED, TACREV, KBP-37 and SemEval-2010 Task8, demonstrate that our approach yields significant improvements over baseline methods. Additionally, we observe that the Att-FMI model exhibits greater robustness against information interference in the extraction of entity relations from longer sentences than previous methods.**

***Index Terms*—relation classification; syntactic dependency parsing; BERT; self-attentive mechanism.**

I. Introduction

ENTITY relation classification is the recognition of the semantic relation between pairs of entities in a sentence, which is one of the most crucial tasks in the field of information extraction [1]. For the following sentence:

" $Ten\ buckets\ of\ [water]_{e1}\ were\ poured\ into\ a\ vacant\ [area]_{e2}\ outside\ the\ house.$ "

where e_1 = "water" and e_2 = "area" are a pair of target entities, identifying the relation "Entity-Destination" between e_1 and e_2 is the purpose of the relation classification task. Since relation extraction plays a significant role in natural language processing (NLP) applications, such as knowledge graph construction [2], machine translation [3] and Automated Q&A [4]. As a result, many researchers have devoted themselves to relational extraction research in the past few years.

Among all the research methods, supervised learning has gradually become dominant in relation extraction tasks. In recent years, massive work related to supervised relation extraction has arisen, and these methods mainly tend to be influenced by the following ideas:

Joint different characteristics. These studies typically combine word embeddings and position features as representations for relation extraction.

Introduce syntactic analysis. To analyze the sequence's grammar and identify the direct connection of target entities, researchers usually require a special parser [5].

Fine-tuning. The approach involves utilizing pre-trained language models (PLMs) [6], such as BERT [7], RoBERTa [8], etc, which are then fine-tuned to the downstream classification task.

Although the aforementioned technique achieves positive outcomes in relation extraction, there still exist limitations. First of all, the majority of studies have not considered the fact that relation classification depends on both the sentence's structure and semantic, not just one alone. Second, some researchers overlook the fact that the weights of semantic and structure information vary according to the sentence itself.

To tackle the issue, we propose a novel fusing multi-information model with a self-attentive strategy (Att-FMI), which consists of a special parser, the pre-trained BERT model and a self-attention mechanism [9-11]. Att-FMI can obtain both the structure representation of sentences with the dependency syntactic parser [12] and semantic representation by using the pre-trained BERT model. In order to make full use of these two representations, Att-FMI is designed automatically adjust the weights subtly of two representations via the self-attentive strategy.

The primary contributions of this article include the following:

(1) We propose a novel Att-FMI model for the relation classification, which are capable of extracting the semantic and structure representations of texts and automatically focusing on the information that contributes to classifying based on the self-attentive strategy.

(2) When handling long texts that contain plenty of irrelevant details, our algorithm still works well, showing

Manuscript received February 18, 2023; revised Jun 26, 2023. This work was supported by the National Natural Science Foundation of China Grant 61762063 and the Research Project of the Education Department of Jiangxi Province Grant GJJ170991.

Xuefeng Fu is an Associate Professor at Nanchang Institute of Technology (NIT), Nanchang 330099, China. (Corresponding author, tel: 86-18170936669, email: fxf@nit.edu.cn)

Kailiang wang is a graduate student of Nanchang Institute of Technology (NIT), Nanchang, China. (email: WKLiang1995@163.com)

Yanping Liu is a graduate student of Nanchang Institute of Technology (NIT), Nanchang, China. (email: 1796164455@qq.com)

Weikun Chen is a graduate student of Nanchang Institute of Technology (NIT), Nanchang, China. (email: 1401619099@qq.com)

Jun Chen is a graduate student of Nanchang Institute of Technology (NIT), Nanchang, China. (email: 1646428688@qq.com)

great robustness to interference.

(3) Without any introduction of external resources and background knowledge, Att-FMI significantly and consistently outperform four popular benchmarks of relation classification, including TACRED, TACREV, KBP-37 and SemEval-2010, compared to existing baselines.

II. Related Work

The existing study of relation classification had gone through three stages including pattern matching [13], machine learning [14] and deep learning [15]. Pattern matching-based approaches mainly included rule-based and word-driven relation extraction. In the early stages of limited specialization and small corpus size, both of them had achieved considerable progress, but they also encountered challenges with poor transferability and the high expense of manual annotation. In machine learning, the two main categories of algorithms were kernel function-based and feature vector-based relation extraction. The strategy significantly outperformed earlier pattern-matching techniques in terms of accuracy, precision and recall. However, it still suffered from the drawbacks of slow computing speed and heavy labour consumption.

Over the years, supervised relation extraction with deep neural networks is gradually becoming a research focus. As opposed to the conventional relation classification methods, it overcomes the key issues of manual feature selection and error propagation of troubling features and contributes to better exploring and utilizing potential information among entities.

To classify relations, Zeng et al. [16] initially employed convolutional neural networks (CNN) to access word-level and sentence-level information while using the softmax layer. This significantly enhanced the task's performance. The shortest dependency path and CNN were coupled for entity relation extraction by Xu et al. [17] at the same time. The performance of this approach had increased when compared to utilizing only CNN, proving that the shortest dependency path is effective for classifying relations.

Socher et al. [18] suggested employing Recurrent neural networks (RNN) to explore entity relation extraction in addition to CNN-based strategies. RNN was implemented in the method to learn a representation of the connections between the target entities in a syntactic parse tree, which was then used for classification. However, there were some challenges such as gradient disappearance and gradient explosion when parsing the sentence by RNN. Researchers began using long short term memory (LSTM) with greater performance for relation extraction as a result.

The application of LSTM to the relation extraction task was presented by Xu et al [19]. This method was based on the shortest path of the syntactic dependency tree and incorporated multiple features for relation classification. Based on the idea that the input at the time was dependent not only on the word preceding it but also on the word following it, Zhang et al. [20] discovered that bi-directional long short-term memory (Bi-LSTM) could capture the bi-directional semantic dependencies to gain additional sequential information. As a consequence, the approach performed better than the model described in the literature [16], proving the value of Bi-LSTM for relation extraction.

An attention mechanism was introduced to decide the crucial information for the relation classification task. Wang et al. [21] proposed to use an attention mechanism combined with CNN or RNN for this work. The entity-aware self-attention mechanism proposed by Yamada et al. [22] considered the types of tokens while computing the attentive fraction and achieved excellent performance on several entity-related tasks.

The pre-trained BERT model launched by Google in 2018 broke performance records across all 11 NLP tasks. With the rise of PLMs, relation extraction researchers have started concentrating more on how to fine-tune PLMs in this work.

Wu et al. [23] tried employing the BERT to handle information from the target entities. Firstly, they located the target entities and transferred the original sequence by using BERT, then concatenated the corresponding encoding of the two entities as a relation representation for classification. In addition, Wang et al. [24] further proposed to introduce external resources based on keeping the original pre-trained model parameters constant.

From the above, we can summarize that the pre-trained BERT model is capable of better understanding the meaning of texts and using the knowledge learned during pre-training for downstream relation classification tasks. However, it also suffers from some flaws. For instance, the pre-trained BERT model is susceptible to interference from noise. The recognition of relational trigger words will be hampered when irregular words, words with logical errors, and misspellings appear in the input sequence. Furthermore, it is unable to learn the grammatical structure of sentences.

Zhao et al. [25] proposed an adaptive learning method for text classification tasks, which fused multiple feature information and enhanced classification performance. Inspired by this, we propose the Att-FMI mdoel, which performs better in the relation extraction task by properly combining semantic and structure information.

III. Methodology

A. Fine-tuning Pre-trained BERT Model

Unlike conventional language models, the pre-trained BERT model is a multi-layer bidirectional transformer encoder. Pre-training is achieved using the masked language model (MLM), which randomly masks some tokens of the input sequence and forecasts the masked token by context.

Given a pre-trained BERT model M, previous fine-tuning methods first convert the instance $x = \{w_1, w_2, ..., w_n\}$ into an input sequence $\{[CLS], w_1, w_2, .., w_n, [SEP]\}$, and use M to encode all tokens of the input sequence into corresponding vectors $\{h_{[CLS]}, h_{w_1}, h_{w_2}, .., h_{w_n}, h_{[SEP]}\}$. For a downstream classification task, a task-specific head is used to compute the probability distribution of label y over the class set Y with the softmax function $p(y|x) = softmax(Wh_{[CLS]+b})$, where $h_{[CLS]}$ is the hidden vector of the special token $[CLS]$, W is a randomly initialized matrix that needs to be optimized and b is a learned bias vector.

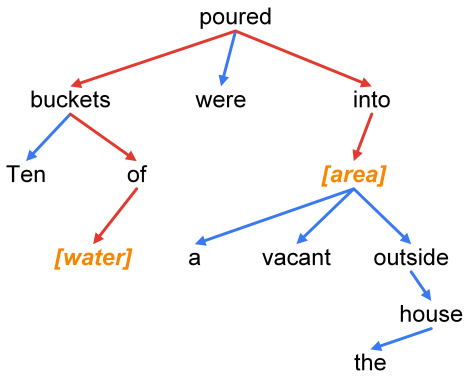


Fig. 1. The dependency tree for the sentence " Ten buckets of water were poured onto a space outside the house. ". The shortest dependency path between subject "$[water]$" and object "$[area]$" is composed of multiple red lines. The edge "$a \rightarrow b$" indicates that a govern b. The dependency type labels between words after parsing are not shown here.

B. Dependency Syntactic Parser

To help relation extraction model relations capture long-range relations between words, a dependency syntactic parser is proposed. The parser can capture the shortest dependency path (SDP) between any two words in a sentence and investigate the complex structure between them, it has been demonstrated to be incredibly successful in relation classification task.

The SDP contains the most critical information while automatically excluding the less relevant words. Ordinarily, dependency structure is displayed in two different ways: either by marking dependency arrows and grammatical relations on the original sentence, or by presenting a dependency tree graph. Figure 1 depicts the dependency parse tree for the given sentence.

C. Overview

The framework for the Att-FMI proposed in this section, which comprises three components, is shown in Figure 2.

(1) **Obtain Semantic Representation**: The pre-trained BERT model encode the input sequence after locating the target entities. To describe the sentence's semantic representation, we then combine the corresponding vector of the target entities with the specific head "[CLS]". (Section E in detail)

(2) **Capture Structure Representation**: By employing dependency syntactic parser, the shortest dependency path between target entities is discovered. The tokens on SDP are then mapped into word embedding and fed into the Bi-LSTM to obtain the last hidden vector as the sentence's structure representation. (Section F in detail)

(3) **Fusion of Two Representations**: Motivated by the attention mechanism, the self-attention strategy is proposed, which can learn the weights of semantic and structure at corpus respectively and automatically focus according to the contribution of two types of information to the classification result. (Section G in detail)

Additionally, we summarize the operational procedure of the Att-FMI model in Algorithm 1 to assist the reader to comprehend the entire process more clearly.

Algorithm 1 : Att-FMI Model

Input: $\mathcal{D} = \{x_n, y_n\}$ // training set
Output: $\hat{\mathcal{Y}} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_n}\}$ // the set of predictive labels

for $n = 1, ..., N$ **do**
  $x_n^{(1)}$, $x_n^{(2)} \leftarrow$ Data Preprocessing for x_n;
  if $x_n^{(1)}$ **then**
    $\{H_0, ..., H_n\} \leftarrow$ BERT-Encoder $(x_n^{(1)})$;
    Calculate entity representations H_{e1}, H_{e2};
    Calculate first token '[CLS]' representations H_{cls};
    $H_{se} \leftarrow$ **Add** $(H_{e1}, H_{e2}, H_{cls})$;
  end if
  if $x_n^{(2)}$ **then**
    $SDP \leftarrow$ Dependency syntactic parser $(x_n^{(2)})$;
    $E_{SDP} \leftarrow$ Word embedding for SDP;
    $H_{st} \leftarrow$ Bi-LSTM (E_{SDP});
  end if
  The embedding layer of BERT: $E_t = \{E_0, ..., E_N\}$;
  Use E_t to calculate average word vector E_s;
  Obtain attention weights Att_{se}, Att_{st} from E_s, H_{se}, H_{st};
  Relation representation M = Concat$(Att_{se} * H_{se}, Att_{st} * H_{st})$;
  Use M to predict result of classification $\hat{y_n}$;
end for
Calculate $Cross - Entropy\ Loss$;
Back propagation and update parameters in Att-FMI model;
return $\hat{\mathcal{Y}} = \{\hat{y_1}, \hat{y_2}, ..., \hat{y_n}\}$;

D. Data Preprocessing

The input sequences are subjected to the following two forms of preprocessing, which assist the model to capture the semantic and structure information.

Before a sentence is encoded by the pre-trained BERT model, we first insert special tokens "[CLS]" and "[SEP]" into the head and tail of the sequence. Meanwhile, in order to make the BERT model can grasp the location information of the two entities accurately, we add the special tokens "$" and "#" at the start and end of the target entities, respectively. For instance, following the addition of the special tokens, the sentence with target entities "water" and "area" will convert to:

" $[CLS]$ *Ten buckets of* \$ *water* \$ *were poured into a vacant* # *area* # *outside the house.* $[SEP]$ "

To strengthen the generality of the model, we also substitute "Entity1" and "Entity2" for the target entities in the sentence before applying the dependency syntactic parser. The specific example is as follows:

" *Ten buckets of Entity*1 *were poured into a vacant Entity*2 *outside the house.* "

E. Sentence Semantic Representation

1) Encoding with The BERT Model: In this paper, we first give an input sequence T with entities $e1$ and $e2$, which outputs the final hidden state output from the BERT model as H, where H_i to H_j refers to the hidden state vector of entity $e1$, H_k to H_m refers to the hidden state vector of entity $e2$, and H_0 corresponds to the hidden state vector of the token "[CLS]" at the beginning of the input sequence. $H_i, H_j, H_k, H_m, H_0 \in R^d$, d is the dimension

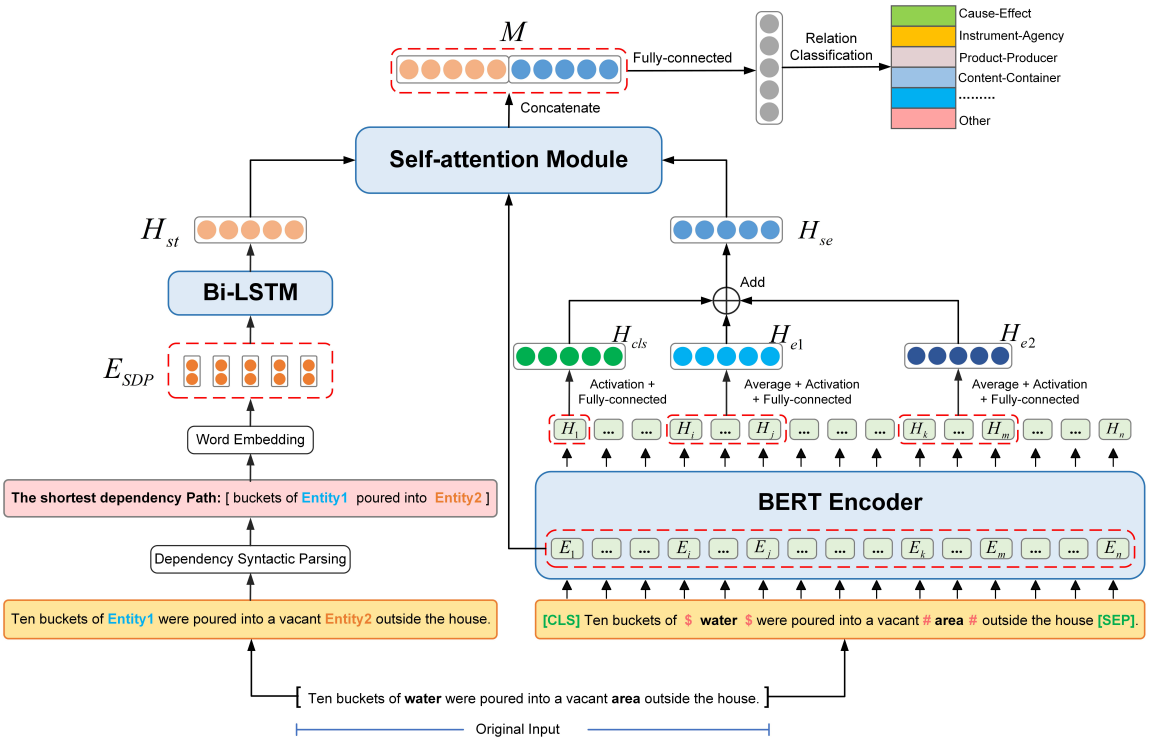


Fig. 2. There exists an overall structure of the Att-FMI model. Once a sentence has been preprocessed, it is fed into the dependency syntactic parser and the BERT model to capture structure and semantic representations, respectively.

of the hidden state vector. The two target entities are given a vector representation by using an average operation, and the results of the activation are then passed to the fully-connected layer. Finally, the vector representations H_{e1} and H_{e2} of entities $e1$ and $e2$ are obtained. Equations (1), (2), and (3) provide a mathematical formulation for this process.

$$H_{e1} = W_1 \times [\tanh(\frac{1}{j-i+1}\sum_{t=i}^{j} H_t)] + b_1 \quad (1)$$

$$H_{e2} = W_2 \times [\tanh(\frac{1}{m-k+1}\sum_{t=k}^{m} H_t)] + b_2 \quad (2)$$

$$H_{cls} = W_0 \times [\tanh(H_0)] + b_0 \quad (3)$$

where $W_0 \in R^{d\times d}$, $W_1 \in R^{d\times d}$, $W_2 \in R^{d\times d}$ are weight matrixes, $b_0 \in R^d$, $b_1 \in R^d$, $b_2 \in R^d$ are biases. We make W_1 and W_2, b_1 and b_2 share the same parameters. That is to say, $W_1 = W_2$, $b_1 = b_2$.

2) Information Integration: The entity representations H_{e1} and H_{e2} contain local features associated with the target entity, while the hidden state vector H_{cls} incorporates the global features of the whole sequence. The three are now fused by adding to generate a semantic representation of the sentence H_{se}:

$$H_{se} = H_{cls} + H_{e1} + H_{e2} \quad (4)$$

where $H_{se} \in R^d$. By merging the local features of the target entities with the global characteristics of the sentence, this method allows H_{se} to integrate contextual semantic information.

F. Sentence Structure Representation

1) Generate The Shortest Dependency Path: When a sentence is so long, the interference information in the sequence increases, resulting in a decrease in the sensitivity of the model to relation trigger words. To tackle the problem, we introduce spaCy [26] (Natural Language Processing Tool) to produce the dependency graph. The shortest distance between the target entities as the starting and ending nodes is SDP:

$$SDP = \{w_{e1}, w_1, w_2 ..., w_n, w_{e2}\} \quad (5)$$

where $w_i (i \in [1, n])$ is a token on the SDP. Most of the fuzz words have been filtered off and only the simplest structure information between the target entities is retained, which is beneficial for the model to identify the most critical relation trigger words in the path.

Firstly, we employ spaCy to parse the sentence and acquire several triples containing dependency type and direction between two words, Table 1 displays the specific outcomes.

According to the results of Table 1, we can construct a dependency graph on the original sentence, where nodes are represented as words on a sequence, and arrows describe the direction of the dependency. As shown in Figure 3, from the dependency graph, we discover the shortest dependency path between the target entities. With eliminating irrelevant words, just the most basic syntactic structure between the target entities is kept.

2) Word Embedding: Due to the temporary lack of a consistent vector representation for dependency type, only the words on the SDP are maintained for embedding.

To map the word w_i on the SDP, we adopt the word vectors produced by GloVe [27], in turn, to the

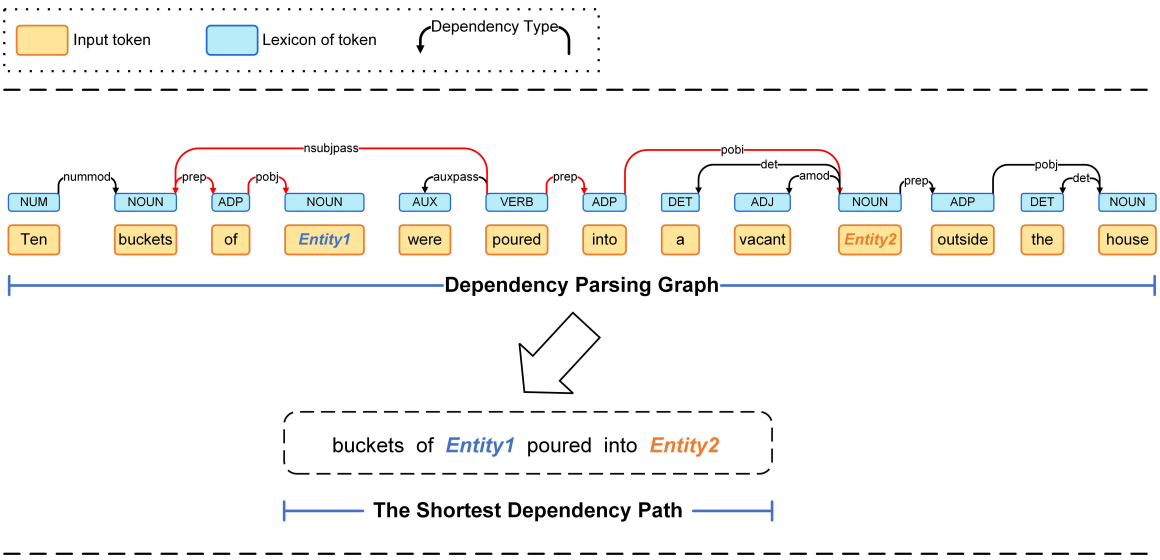


Fig. 3. This diagram illustrates how to construct the shortest dependency path. In the dependency parsing graph, the words, their corresponding lexical properties and the dependency types of the words are shown. By linking between "$Entity1$" and "$Entity2$", we can find the shortest dependency path between them.

TABLE I
THE RESULT OF THE DEPENDENCY PARSING IS THIS, WHERE THE DEPENDENCY TYPE ILLUSTRATES THE RELATION BETWEEN THE TWO LEXICONS AND "→" REFLECTS THE SUBJECT AND OBJECT OF THE DEPENDENCY.

Dependency Type	Direction
nsubjpass	poured→buckets
auxpass	poured→were
prep	poured→into
nummod	buckets→Ten
prep	buckets→of
pobj	into→Entity2
pobj	of→Entity1
det	Entity2→a
amod	Entity2→vacant
prep	Entity2→outside
pobj	outside→house
det	house→the

corresponding word vectors e_{w_i}:

$$e_{w_i} = E(w_i) \tag{6}$$

where the word vector dimension of e_{w_i} is l_w, the SDP can be denoted as the corresponding embedding matrix E_{SDP}:

$$E_{SDP} = [e_{w_1}, e_{w_2}, .., e_{w_n}] \tag{7}$$

where $E_{SDP} \in R^{n \times l_w}$, n is the length of the SDP.

3) Encoding with Bi-LSTM: while processing long-term sequences, RNN experiences gradient disappearance and explosion. To tackle this issue, we employ a bi-directional long short-term memory network (Bi-LSTM) to encode the word embedding matrix E_{SDP}, which effectively captures more comprehensive sequence features.

By using the Bi-LSTM network to encode the word embedding matrix E_{SDP}, we can obtain the final hidden state output H_{st}, which is the sentence's structure representation:

$$H_{st} = h_{SDP} = Bi - LSTM(E_{SDP}) \tag{8}$$

where $H_{st} \in R^d$, d is the dimension of the hidden layer in the Bi-LSTM network.

G. Self-Attention Strategy

In common, the weights of semantic and structure information vary according to the sentence itself, hence the necessity of an efficient way to integrate the two pieces of information. As shown in Figure 4, motivated by the attention mechanism, we propose the self-attention strategy to learn weights at the instance level. The method can compute the attention weights of the structure representation H_{st} and the semantic representation H_{se} respectively, and fuse them according to the attention weights.

First, we apply the average operation on word embeddings of the sentence to get efficient sentence representation E_s, which contains extensive contextual information:

$$E_s = \frac{1}{p+1} \sum_{t=0}^{p} E_t \tag{9}$$

where E_0 to E_p correspond to the word embedding of the sequence, encoded by the embedding layer of BERT respectively. Second, the fully-connected layer is added to E_s to convert the representations into semantic space output E_{s1} and structure space output E_{s2}, which is formally expressed as:

$$E_{s1} = W_3 \times E_s + b_3 \tag{10}$$

$$E_{s2} = W_4 \times E_s + b_4 \tag{11}$$

where $W_3 \in R^{d \times d}$, $W_4 \in R^{d \times d}$ are weight matrixes, $b_3 \in R^d$, $b_4 \in R^d$ are biases, $E_{s1} \in R^d$, $E_{s2} \in R^d$. Third, we capture the similarity of representations by computing the inner product. The output p_{se} and p_{st} are from H_{se}

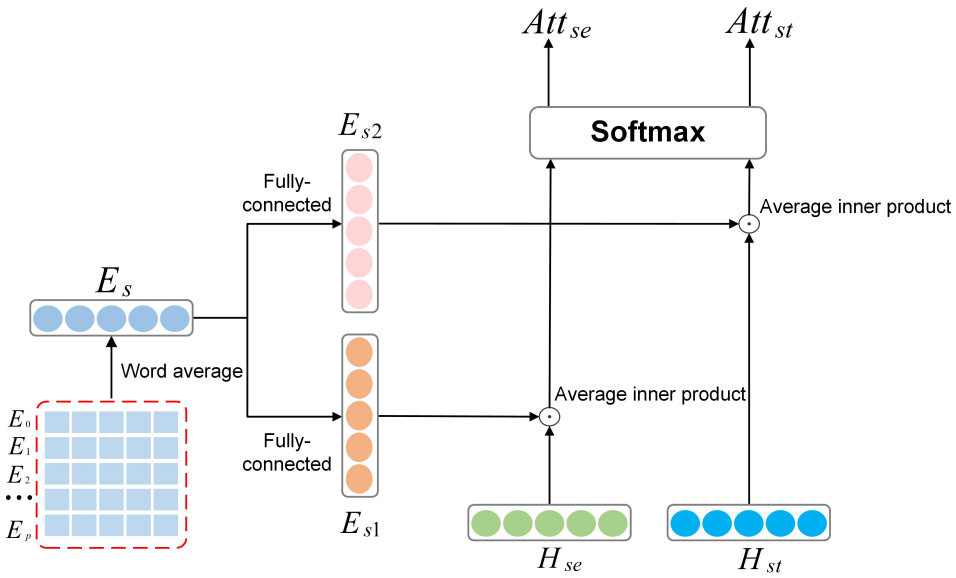


Fig. 4. The self-attention module illustrates how self-attention weights are constructed for two representations.

and H_{st} individually. The specific calculation process is formalized as equations (12) and (13).

$$p_{se} = E_{s1} \odot H_{se} \tag{12}$$

$$p_{st} = E_{s2} \odot H_{st} \tag{13}$$

Last but not least, we use $softmax$ to normalize weights.

$$(Att_{se}, Att_{st}) = softmax(p_{se}, p_{st}) \tag{14}$$

where Att_{se} and Att_{st} are the attention weights normalized, which denote the contribution of the semantic representation H_{se} and the structure representation H_{st} to the final classification results, respectively. According to the weights, we reasonably concatenate H_{se} and H_{st} as the final relation representation M.

$$M = Concat(Att_{se} * H_{se}, Att_{st} * H_{st}) \tag{15}$$

where $M \in R^{2d}$. In summary, the self-attention strategy help model learn the weights of the two representations and effectively combine semantic information and structure information to enhance the performance of the relation extraction task.

H. Model Training and Output

After the above operations are completed, we utilize a softmax-classifier to predict the relation label $\hat{y}$. The classifier takes the relation representation M as input:

$$\tilde{M} = W_6 \times [\tanh(W_5 \times M + b_5)] + b_6 \tag{16}$$

$$p = softmax(\tilde{M}) \tag{17}$$

where $W_5 \in R^{2d \times d}$, $W_6 \in R^{d \times L}$ are weight matrixes, $b_5 \in R^d$, $b_6 \in R^L$ are biases, L refers to the number of relation types within the dataset. According to the probability-distribution p, we can predict the relation of the entity pair e_1 and e_2:

$$\hat{y} = argmax(p) \tag{18}$$

Then, we use Cross-Entropy serves as the loss function for the Att-FMI model:

$$Loss = -\frac{1}{N} \sum_{n=1}^{N} y_n \times \log(\hat{y_n}) \tag{19}$$

Where y_n and $\hat{y_n}$ represent true and predicted labels respectively. The training set has N instances in total.

Adam is an optimizer for the aforementioned procedure to minimize cross-entropy loss. The parameters of our model are randomly initialized and then update by employing back-propagation. Meanwhile, in order to alleviate the overfitting, the Att-FMI model randomly discards some neurons in the neural network by employing the dropout layer.

IV. Experiment

In this section, to demonstrate the effectiveness of the model, we conduct experiments on four commonly used relation classification datasets.

A. Datasets and Metric

TACRED [28]: one of the largest and most widely used datasets for relation classification. It is obtained via crowd-sourcing and contains 42 relation types (including "no_relation").

TACREV [29]: A dataset built on the original TACRED. The errors in the original TACRED development and test sets are corrected while the training set remains unchanged, with the same number of samples and relation types.

KBP-37 [30]: in this dataset, there are 18 types of directed entity relations and one no-relationship type "no_relation", for a total of 37 relation categories.

SemEval-2010 Task8 (SemEval) [31]: an established dataset for relation classification, which involves 9 types

TABLE II
STATISTICS OF RE DATASETS.

Dataset	#Train	#Dev	#Test	#Rel	#Ratio(%)
SemEval	6,507	1,493	2,717	19	14.3
KBP-37	15,917	1,724	3,405	37	78.1
TACRED	68,124	22,631	15,509	42	87.4
TACREV	68,124	22,631	15,509	42	87.4

of directional relationships, for example, the relation "Component-Whole (e1, e2)" and "Component-Whole (e2, e1)" are treated as two different types and one special type "Other".

As shown in Table 2, the distinction between the four datasets is reflected in the number of examples and relation categories and the proportion of long sentences (sentence length $\geq$ 20).

Compared with the SemEval-2010 Task8, it is quite obvious that the KBP-37, TACRED and TACREV are capable of better validating the model's resistance to information interference due to its multiple relationship types and a higher proportion of long sentences. For all the above datasets, F_1 scores serve as our primary standard for evaluation.

B. Experimental Detials

Depending on the size of the datasets, we encoded the sample from TACRED and TACREV using RoBERTa-Large as a pre-trained language model. For samples in the dataset SemEval and KBP-37, we use RoBERTa-Base to encode. In the meanwhile, the hidden size in the Bi-LSTM network must match the hidden size in the RoBERTa model in the experiments. Most hyper-parameters are set following previous works. The key settings in our testing are shown in Table 3:

TABLE III
PARAMETER SETTINGS.

Parameters	Value
Number of epochs	30
Batch size	16
Max sequence length	128
Learning rate	3e-5
Hidden size	768
Word dim	200
Dropout	0.3

C. Baseline Methods

To verify the performance of the proposed Att-FMI model, we compared experimental results with a number of baseline method from recent years.

BiLSTM-CNN [32]: The method employed Bi-LSTM to extract long-range relations between labels to produce higher-level semantic representation, which were then fed to CNN for relation classification, combining the strengths of RNN and CNN.

PA-LSTM [28]: For the relation extraction challenge, a neural sequence model of cognitive location is put forth that can fully incorporate information about semantic similarity and location based on the attention mechanism.

GLFN [33]: Word temporal properties were obtained using a recurrent neural network, and they were further divided into local and global temporal features using a convolutional neural network. In the end, the model could combine and filter the two aspects to produce a thorough representation of the sentence semantics after introducing a self-attention mechanism.

C-GCN [34]: The paper proposed an idea using a new pruning strategy for noisy data filtering and introducing graph convolutional networks to relation extraction tasks to convolve the syntactic dependency graph of the text.

ATT-Gate-GCN [35]: An attention-guided gate-aware graph convolution model based on attention was proposed. While using the attention mechanism to perform soft pruning, a graph convolutional network was constructed to obtain relational features, and the two were integrated for extracting crucial information.

R-BERT [23]: Based on BERT, a relational representation for classifying was generated by combining the information from the sentence and the target entity.

Know-BERT [36]: In order to create a knowledge-enhanced entity span representation, the approach was proposed to explicitly model entity span in the input sequences by incorporating external information and jointly training entity connectors. The contextual word representation was updated in the meanwhile to make sure it had all of the entity information using the word-to-entity attention mechanism.

MTB [37]: A novel pre-training task matching the blanks was launched using BERT, which learned the representation of relations from text without the use of a knowledge graph or human annotation supervision.

Span-BERT [38]: A new pre-training method at the word level is suggested that uses a representation of the word boundaries to forecast the content of the location to which the [MASK] is added.

LUKE [39]: The researcher suggests an entity-aware self-attentive mechanism and a job specifically for dealing with entity-related contextual representation in a large text corpus and knowledge network for pre-training.

D. Comparison with Other Methods

The test results in Table 4 reveal that the approaches using the pre-trained model significantly exceed the methods

TABLE IV
COMPARISON WITH RESULTS IN THE LITERATURE F_1 SCORES (%). "-" INDICATES EXPERIMENTAL RESULTS ARE NOT PUBLISHED WITH THE SOURCE PAPER AND THE REST OF THE RESULTS ARE FROM THE ORIGINAL PAPER. "W/O" MEANS THAT NO ADDITIONAL DATA IS USED FOR PRE-TRAINING AND FINE-TUNING, YET "W/" MEANS THAT EXTRA DATA ARE USING FOR DATA AUGMENTATION. THE BEST RESULTS ARE BOLD.

Method	Extra Data	TACRED	TACREV	KBP-37	SemEval
Based on CNN/RNN					
BiLSTM-CNN [32]	w/o	-	-	60.1	81.9
PA-LSTM [28]	w/o	65.1	73.3	-	84.8
GLFN [33]	w/o	-	-	64.9	86.1
Based on GNN					
C-GCN [34]	w/o	66.3	74.6	-	84.8
Att-Gate-GCN [35]	w/o	-	-	61.7	85.9
Based on PLMs					
R-BERT [23]	w/o	-	-	-	89.25
Know-BERT [36]	w/	71.5	79.3	-	89.1
MTB [37]	w/	70.1	-	69.3	89.5
Span-BERT [38]	w/	70.8	78.0	-	-
LUKE [39]	w/	72.7	80.6	-	-
Att-FMI	w/o	**75.5**	**83.8**	**71.4**	**90.14**

based on CNN/RNN or GNN in performance. We believe that pre-trained models are more effective in enhancing the performance of relation extraction tasks than traditional models.

As opposed to similar approaches based on PLMs, our Att-FMI model outperforms all other models in terms of F_1 scores. On the four datasets TARCED, TACREV, KBP-37 and SemEval, our Att-FMI model exceeds the best baseline with a relative improvement of 2.8%, 3.2%, 2.1% and 0.61%, respectively.

E. Analysis of Model Performance

Considering the difficulty of model training and the cost of complex manual design, we compared the performance of our Att-FMI model with other baseline methods based on PLMs in terms of both the number of parameters and the use of manually labelled external knowledge, as shown in Table 3.

Although KnowBERT and MTB employ BERT-Base as the language model, allowing for a minimal number of parameters, they each integrate a substantial quantity of manually labelled external knowledge as data augmentation throughout the fine-tuning and pre-training phase. Contrarily, the Att-FMI model achieves significantly better F_1 scores than existing models on all four datasets without the addition of any extraneous knowledge or difficult manual design conditions, proving the approach's effectiveness and conciseness as well as its strong potential for relational extraction tasks.

F. Choice of Contextual Representation

In the self-attention strategy, in order to assist the model learn through the corpus and obtain attention weights of semantic and structure representations, we require a special feature vector (E_s in Section 3.3) incorporating contextual information, which originates from the mean of the word embeddings.

TABLE V
COMPARISON OF MODEL REQUIREMENTS.

Method	Pre-trained Model	Size	Extra data
R-BERT	BERT-Large	340M	w/o
Know-BERT	BERT-Base	110M	w/
MTB	BERT-Base	110M	w/
Span-BERT	BERT-Large	340M	w/
LUCK	LUKE-500K-Large	480M	w/
Att-FMI(TACRED)	RoBERTa-Large	340M	w/o
Att-FMI(TACREV)	RoBERTa-Large	340M	w/o
Att-FMI(KBP-37)	RoBERTa-Base	110M	w/o
Att-FMI(SemEval)	RoBERTa-Base	110M	w/o

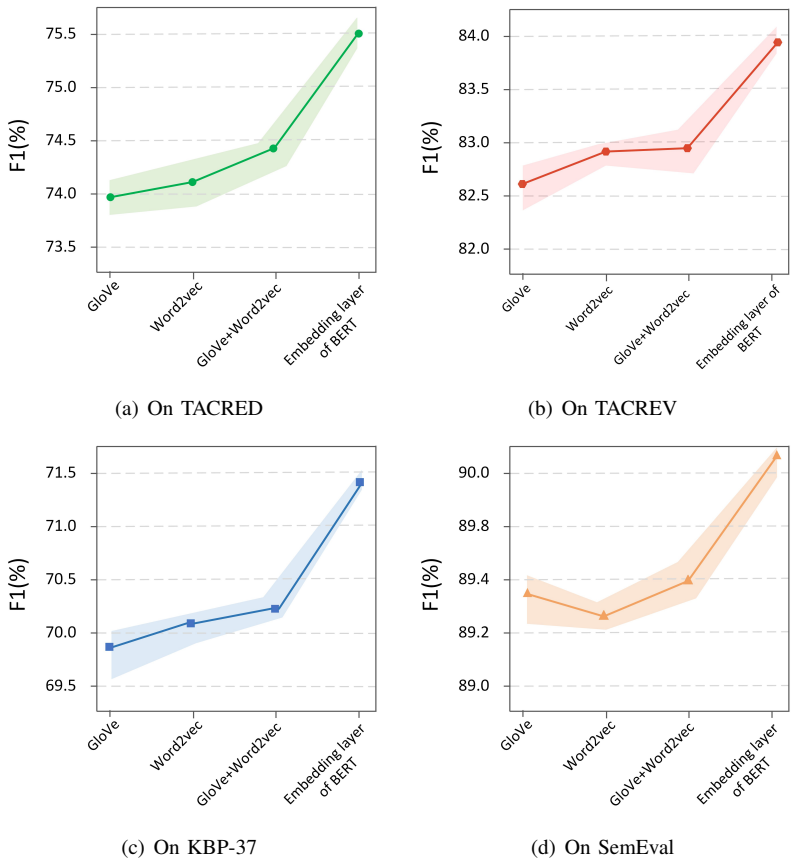
There are four existing word embedding methods: GloVe, Word2vec [40], Word2vec + GloVe and the embedding layer of the BERT model. We choose the last one for our approach. Figure 5 illustrates the comparison of the implementation of different word embedding.

Performance is the best on all datasets with the embedding layer of the BERT model. We consider that is because the embedding layer of BERT is the composition of "Token Embedding", "Segment Embedding" and "Position Embedding" of the corresponding token of the input sequence, compared with other embedding methods, which can incorporate more information related to the token.

G. Ablation Study

Empirical results have demonstrated the viability of our approach. Next, we further want to understand the specific contributions of the components of the Att-FMI model. For this purpose, we create three more configurations and evaluate their performance by $Precision$, $Recall$ and F_1 scores on different datasets.

Att-FMI-NO-ATT: The first configuration is to drop the self-attention strategy and directly combine semantic and



(a) On TACRED (b) On TACREV

(c) On KBP-37 (d) On SemEval

Fig. 5. F_1 scores with various word embedding methods on RE datasets.

structure representations to get the relation representation for classification. That is to say, this method gives equal weights to these two kinds of representations. Meanwhile, keep the rest settings the same.

Att-FMI-NO-SDP: The second configuration is to discard the shortest path dependency of the input sequence during the forward. In other words, we use the Bi-LSTM network to encode a sentence to obtain the structure representation. The other settings stay the same.

Att-FMI-NO-ENT: The third configuration is to remove the hidden vector output of two entities from the semantic representation. That is, only H_{cls} as semantic representation is directly fed into the self-attentive module. Simultaneously, the other settings hold the same.

The results of the ablation study with the above three configurations are shown in Table 6. We find that the Att-FMI model has achieved leading performance in terms of $Accuracy$, $Recall$ and F_1 scores on all datasets compared to the other configurations. Now, we make a specific analysis for each configuration:

(1) With the condition of dropping the self-attention strategy, we observe that the performance of the model is slightly decreased, which proves that the self-attention mechanism plays an essential role in the efficient fusion of the semantic and structure information.

(2) If we discard the use of dependency syntactic parser, experimental results would be much worse, indicating that the parser indeed contributes to capturing the structure information of the sentence. Simultaneously, by comparing the model's performance on the four benchmark datasets cross-sectionally, we further find that $Precision$, $Recall$ and F_1 scores all decline more on the datasets KBP-37, TACRED and TACREV compared to the SemEval dataset. The analysis of the datasets in "A. Datasets and Metric" of this chapter shows that longer sentences are more abundant in the latter three datasets. Therefore, we argue that the shortest dependency path is favourable to grasping the structure information of long sentences and minimizes the

TABLE VI
COMPARSION OF PRECISION, RECALL AND F_1 SCORES OF METHODS WITH DIFFERENT COMPONENTS. THE BEST RESULTS ARE BOLD.

Configuration	TACRED			TACREV			KBP-37			SemEval		
	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$
Att-FMI-NO-ATT	73.39	**75.69**	74.52	82.11	82.79	82.45	68.8	70.28	69.53	88.84	90.33	89.58
Att-FMI-NO-SDP	72.1	73.57	72.83	80.04	82.2	81.11	68.05	68.98	68.51	**90.04**	88.4	89.21
Att-FMI-NO-ENT	71.96	70.87	71.41	80.45	79.02	79.73	66.67	69.03	67.83	86.92	88.88	87.89
Att-FMI	**76.1**	74.91	**75.5**	**82.4**	**85.25**	**83.8**	**69.97**	**72.89**	**71.4**	89.54	**90.75**	**90.14**

interference of irrelevant words.

(3) Without considering the hidden vector output of the two target entities, the performance of the approach is the worst among the three configurations, which reflects that integrating the information of the target entity assists in enhancing the semantic representation.

V. Conclusion

In this paper, we propose a fusing multi-information model with a self-attentive strategy (Att-FMI) for many-class relation classification by capturing and fusing both semantic and structure information effectively. The experimental results on four benchmark relation classification datasets show that Att-FMI significantly outperforms the baselines without manual annotations and the introduction of extra knowledge. Meanwhile, our model also shows boosted resistance to interference information than previous methods.

In the future, we plan to examine ways to apply pre-trained language models and dependency syntactic parser to joint entity-relation extraction tasks, with the aim of further enhancing our model's performance on multiple relation extraction tasks.

References

[1] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A Novel Cascade Binary Tagging Framework for Relational Triple Extraction," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online, pp.1476-1488, 2020.

[2] Z. Lin, D. Yang, H. Jiang, and H. Yin, "Learning Patient Similarity via Heterogeneous Medical Knowledge Graph Embedding," in *IAENG International Journal of Computer Science*, vol. 48, no. 4, pp.868-877, 2021.

[3] Y. Qiu, Y. Wang, L. Bai, Z. Yin, H. Shen, and S. Bai, "Multi-coverage Model for Neural Machine Translation," *Chinese Journal of Electronics*, vol. 50, no. 09, pp.2242-2264, 2022.

[4] T. G. Soares, A. Azhari, N. Rokhman and E. Winarko, "Education Question Answering Systems: A Survey," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2021*, Hong Kong, pp.24-34, 2021.

[5] R. Qing, K. Chen and R. Pang, "Research on Traditional Mongolian-Chinese Neural Machine Translation Based on Dependency Syntactic Information and Transformer Model," *Applied Science*, vol. 12, no. 19, pp.1074, 2022.

[6] G. L. He, C. Y. Chi and Y. Y. Zhan, "Combining N-gram Statistical Model with Pre-trained Model to Correct Chinese Sentence Error," in *Engineering Letters* vol. 30, no. 2, pp.476-484, 2022.

[7] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in *Proceeding of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, Minneapolis, USA, pp.4171-4186, 2019.

[8] A. Vaswani, N. Shazeer, and N. Parmar, "RoBERTa: A Robustly Optimized BERT Pretraining Approach," in *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*, los Angeles, UAS, pp.6000-6010, 2017.

[9] X. Zhang, G. Yu, J. Shang, and B. Zhang, "Short-term Traffic Flow Prediction With Residual Graph Attention Network," in *Engineering Letters* vol. 30, no. 4, pp.1230-1236, 2022.

[10] Y. Wang, X. Cheng, and X. Meng, "Sentiment Analysis with An Integrated Model of BERT and Bi-LSTM Based on Mult-Head Attention Mechanism," in *IAENG International Journal of Computer Science*, vol. 50, no. 1, pp.255-262, 2023.

[11] X. Yu, Z. Li, J Wu, and M. Liu, "Multi-module Fusion Relevance Attention Network for Multi-label Text Classification," in *Engineering Letters*, vol. 30, no. 4, pp.1237-1245, 2022.

[12] D. Li, L. Yan, J. Yang, and Z. Ma, "Dependency syntax guided BERT-BiLSTM-GAM-CRF for Chinese NER," *Expert Systems with Applications*, vol. 196, pp.116682, 2022.

[13] M. Ueda, Y. Matsunami, P. Siriaraya, and S. Nakajima, "Developing Evaluation Expression Dictionaries for the Cosmetic Review Recommendation," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2019*, Hong Kong, pp.236-241, 2019.

[14] A. A. Syed, Y. H. Lukas, and A. Wibowo, "A Comparision of Machine Learning Classifiers on Laptop Products Classification Task," in *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2021*, Hong Kong, pp.104-110, 2021.

[15] D. Li, Y. Zhang, Y. Li, and D. Lin, "A Review of Entity Relation Extraction Methods," *Journal of Computer Research and Development*, vol. 57, no. 07, pp.1424-1448, 2020.

[16] D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao, "Relation Classification via Convolutional Deep Neural Network," in *Proceedings of the 25th International Conference on Computational Linguistics: Technical Paper (COLING)*, Dublin, Ireland, pp.2335-2344, 2014.

[17] K. Xu, Y. Feng, S. Huang, and D. Zhao, "Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling," *arXiv 2015*, arXiv:1506.07650, 2019.

[18] R. Socher, B. Huval, C. Manning, and C. Ng, "Semantic compositionality through recursive matrix-vector spaces," in *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, Jeju Island, Korea, pp.1201-1211, 2012.

[19] Y. Xu, L. Mou, G. Li, Y. Chen and H. Peng, "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 (EMNLP)*, Lisbon, Portugal, pp.1785-1789, 2015.

[20] S. Zhang, D. Zheng, X. Hu, and M. Yang, "Bidirectional Long Short-Term Memory Networks for Relation Classification," in *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation (PACLIC)*, Shanghai, China, pp.73-78, 2015.

[21] L. Wang, Z. Gao, M. De, and Z. Liu, "Relation Classification via Multi-Level Attention CNNs," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL)*, Berlin, Germany, pp.1298-1307, 2016.

[22] I. Yamada, A. Asai, H. Shindo, H. Takeda and M. Yuji, "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp.6442-6454, 2020.

[23] S. Wu, Y. He, "Enriching Pre-trained Language Model with Entity Information for Relation Classification," in *Proceedings of the 28th ACM International Conference on Information and Knowledge Management (CIKM)*, Bejing, China, pp.2361-2364, 2019.

[24] R. Wang, D. Tang, N. Duan, Z, Wei and X. Huang, "K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters," *arXiv 2020*, arXiv:2002.01808, 2020.

[25] J. Zhao, Z. Zhan, Q. Yang, Y. Zhang, and C. Hu, "Adaptive Learning of Local Semantic and Global Structure Representations for Text Classification," in *Proceedings the 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, New Mexico, USA, pp.2033-2043, 2018.

[26] spaCy. Available Online: https://spacy.io/ (accessed on 7 May 2023).

[27] J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp.1532-1543, 2014.

[28] Y. Zhang, V. Zhong, D. Chen, G. Angeli, and C. D. Manning, "Position-aware attention and supervised data improve slot filling," in *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Denmark, pp.34-45, 2017.

[29] C. Alt, A. Gabryszak, and L, Hennig, "TACRED Revisited: A Thorough Evaluation of the TACRED Relation Extraction Task," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, Online, pp.1558-1569, 2020.

[30] G. Angeli, J. Tibshirani, J. Wu, and C.-D. Manning, "Combining Distant and Partial Supervision for Relation Extraction," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar, pp.1556–1567, 2014.

[31] I. Hendrickx, S.-N. Kim, Z. Kozareva, and Z. Nakov, "SemEval-2010 Task 8: Multi-Way Classification of Semantic Relations Between Pairs of Nominals," *arXiv 2019*, arXiv:1911.10422, 2019.

[32] L. Zhang, F. Xiang, "Relation Classification via BiLSTM-CNN," in *Proceedings of the 3th International Conference on Data Mining and Big Data (DMBD)*, Shanghai, China, pp.373-382, 2018.

[33] W. Song, and F. Zhou, "Relation Extraction Method based on Global and Local Feature-aware Networks," *Journal of Chinese Information Processing*, vol. 34, no. 11, pp.96-103, 2020.
[34] Y, Zhang, P. Qi, and C.-D. Manning, "Graph Convolution over Pruned Dependency Trees Improves Relation Extraction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Brussels, Belgium, pp.2205-2215, 2018.
[35] X. Wang, X. Qian, and W. Song, "A Relation Extraction Model with Attention and Graph Convolutional Networks," *Journal of Computer Applications*, vol. 41, no. 02, pp.350-356, 2021.
[36] M.-E. Peters, M. Neumann, R.-L. Logan IV, R. Schwartz, and V. Joshi, "Knowledge Enhanced Contextual Word Representations," *arXiv 2019*, arXiv:1909.04164, 2019.
[37] L.-B. Soares, N. Fitzgerald, J. Ling, and K. Tom, "Matching the Blanks: Distributional Similarity for Relation Learning," in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (ACL)*, Florence, Italy, pp.2895-2905, 2019.
[38] M. Joshi, D. Chen, Y. Liu, D. S. Weld, L. Zettlemoyer, and O. Levy, "SpanBERT: Improving Pre-training by Representing and Predicting Spans," in *Transactions of the Association for Computational Linguistics 8*, MA, pp.64-77, 2020.
[39] I. Yamada, A. Asai, H. Shindo, H. Takeda, and Y. Matsumoto, "LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, pp.5442-6454, 2020.
[40] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *arXiv 2013*, arXiv:1301.3781, 2013.

Xuefeng Fu received Ph.D. degree from Southeast University, China, in 2016. His major research areas are deep learning-based knowledge graph construction and inference. He is specialized in symbolic logic for representing and reasoning about imprecise knowledge.

He works as an associate professor at the Nanchang Institute of technology (NIT) since 2005. He is in charge of the National Natural Science Foundation of China project "Research on efficient non-standard reasoning techniques for graph-based description logic" and presided over the project of Jiangxi Provincial Natural Science Foundation "Research on OWL ontology debugging method based on the graph". He is also a senior member of the CCF (China Computer Federation).

Kailiang Wang is a graduate student at the Nanchang Institute of Technology (NIT). His main research interests include machine learning and information extraction.

Yanping Liu is a graduate student at the Nanchang Institute of Technology (NIT). Her main research interest is multiple sentiment analysis.

Weikun Chen is a graduate student at Nanchang Institute of Technology (NIT). His main research interests include knowledge graphs and recommend systems.

Jun Chen is a graduate student of Nanchang Institute of Technology (NIT). His main research interests include deep learning and recommend the system.