Neural Computing and Applications (2022) 34:3471–3481
https://doi.org/10.1007/s00521-021-05815-z

SPECIAL ISSUE ON MULTI-MODAL INFORMATION LEARNING AND ANALYTICS ON BIG DATA

A joint model for entity and relation extraction based on BERT

Bo Qiao[1] · Zhuoyang Zou[1] · Yu Huang[2] · Kui Fang[1] · Xinghui Zhu[1] · Yiming Chen[1]

Received: 16 October 2020 / Accepted: 9 February 2021 / Published online: 8 March 2021

Abstract

In recent years, as the knowledge graph has attained significant achievements in many specific fields, which has become one of the core driving forces for the development of the internet and artificial intelligence. However, there is no mature knowledge graph in the field of agriculture, so it is a great significance study on the construction technology of agricultural knowledge graph. Named entity recognition and relation extraction are key steps in the construction of knowledge graph. In this paper, based on the joint extraction model LSTM-LSTM-Bias brought in BERT pre-training language model to proposed a agricultural entity relationship joint extraction model BERT-BILSTM-LSTM which is applied to the standard data set NYT and self-built agricultural data set AgriRelation. Experimental results showed that the model can effectively extracted the relationship between agricultural entities and entities.

Keywords Agricultural knowledge graph · Named entity recognition · Relation extraction · Joint extraction · BERT

1 Introduction

Information extraction (IE) is the first step in the construction of knowledge graphs, which is to convert unstructured or semi-structured natural language text into structured data. Named entity recognition (NER) and relation extraction (RE) are two important subtasks of IE. The purpose of NER is to identify entity information with special or referential significance from the text, and RE is responsible for extracting the entity semantic relationship from the text and getting the entity–relation triples like <entity1, relationship, entity2>.

The traditional pipeline methods treat entity extraction and relation extraction as two independent processes. After identified the entities in the sentences, the follow task is to identify entities combined in pairs and then classified its relationship. The pipeline methods are relatively simple in modelling, but the correlation between the two subtasks is not considered in the training process. They are easy to cause error propagation, and the errors of the entity recognition task will affect the performance of subsequent relationship classification. In addition, unrelated entities will bring redundant information, thereby increasing the error rate.

In recent years, many works have considered the joint modeling of entity recognition and relationship extraction tasks. These end-to-end models have also brought significantly better results. However, the existing joint extraction models use static word vector representation for word embedding, and do not take into account that the same word may have different semantics, and cannot model polysemous words. To solve this problem, we replaced the

✉ Kui Fang
fk@hunau.edu.cn

✉ Xinghui Zhu
zhuxh@hunau.edu.cn

Bo Qiao
qiaobo@hunau.edu.cn

Zhuoyang Zou
zzy@stu.hunau.edu.cn

Yu Huang
huangyu@hunau.edu.cn

Yiming Chen
chenym@hunau.edu.cn

[1] College of Information and Intelligence, Hunan Agricultural University, Changsha, Hunan, China

[2] College of Biological Science and Technology, Hunan Agricultural University, Changsha, Hunan, China

static word embedding [1] in the LSTM-LSTM-Bias model proposed by Zheng et al. [2] with a dynamic fine-tuning method [3] to solve downstream tasks. Our model effectively solved the problem that the original model cannot model polysemous words.

The main contributions of this paper are as follows:

1. We have improved the joint extraction model of Zheng et al. [2], which currently has excellent results. We introduced the pre-training language model BERT on the basis of their model [4] and proposed a joint extraction model BERT-BILSTM-LSTM. The model achieved an F1 score of 55.9% on the NYT standard data set, which is 3.9 percentage points higher than the result of Zheng et al.
2. We constructed the agricultural data set AgriRelation, and used the BERT-BILSTM-LSTM model to extract relation, and obtained a $F1$ score of 57.6%. It is verified that the model can also extract entity relations when the sample data set is small.

The rest parts of this paper are organized as follows: Sect. 2 briefly introduces relevant works, Sect. 3 comes up with the BERT-BILSTM-LSTM model, and Sect. 4 states the environment, data, parameter settings and results relating to the experiments with the model. And in the final, the conclusion based on above works is given in Sect. 5.

2 Related work

2.1 Named entity recognition

Entity is an important language unit that carries information in the text. A fundamental semantic expression can be expressed as the entities that contains and the association and interaction among these entities. Entities are also the core units of knowledge graph. Knowledge graph is usually a huge knowledge network with entities as nodes. Named entity recognition refers to the task of recognizing named entities in the text and classifying them into designated categories, which is the basis for understanding the meaning of text. NER technology can detect new entities in the text and add them to the existing database. It is the core technology of knowledge graph construction.

Since the 1990s, statistical models have been the mainstream method of entity recognition. There are many statistical methods used to extract entities in text, such as hidden Markov model [5, 6], Maximum Entropy model [7, 8] and Support Vector Machines [9]. However, traditional statistical models require a large amount of annotated corpus to learn information, which leads to the bottleneck of constructing information extraction system in open domain or Web environment. With the popularity of deep learning in different fields, more and more deep learning models are proposed to solve entity recognition problems [10–14].

2.2 Relation extraction

Entity relationship describes the association relationship of existing things, and it is defined as a certain connection for two or more entities, which is the basis for the automatic construction of knowledge graph and natural language understanding. Relation extraction is to automatically detect and identify a certain semantic relationship between entities from the text. It systematically processes various unstructured/semi-structured text inputs (such as news pages, product pages, Weibo, forum pages), using a variety of technologies to identify and discover the relationship between various predefined categories and open categories, which has important theoretical significance and broad application prospects to provide a variety of applications important support.

Relation extraction has been continuously studied in the past two decades. Feature engineering [15], kernel methods [16, 17], and graph models [18] have been widely used in them, and some results have been achieved. With the advent of the deep learning era, neural network models have brought new breakthroughs in relation extraction. In 2014, Zeng et al. [19] improved the accuracy of the relationship extraction model by extracting the features of word level and sentence level with CNN and classifying the relationship by combing the hidden layer and softmax layer. Nguyen and Grishman [20] improved on Zeng's work by adding a multi-size convolution kernel and extracting the characteristics of sentences level. Santos et al. modified the loss function used in Zeng's model into a new pairwise ranking loss function [21]. Considering the unsatisfactory modeling effect of CNN for long distance text sequences, Socher et al. took the lead in using RNN for entity relationship extraction [22]. Zhou et al. [23] combined attention and BiLSTM to conduct the experiment of relationship classification. Lin et al. [24] proposed a self-training framework and built a recursive neural network embedded with multiple semantic isomeric elements within the framework. Zhang et al. [25] proposed an extended graph convolutional neural network, which can effectively process arbitrary-dependent structures in parallel and facilitate the extraction of entity relations. Zhu et al. [26] proposed a method to generate graph neural network parameters based on natural language statements to enable the neural network to perform relational reasoning on unstructured text input. In addition, BERT is being used in more and more relational extraction models for pre-training. Shi and Lin [27] proposed a simple model based on BERT, which can be used for relationship extraction and semantic role annotation. Shen et al. [28] used BERT to extract the relationship between characters, reducing the impact of noise data on the relation extraction model.

2.3 Joint extraction

The term joint learning is not a term that has only recently appeared. In the field of natural language processing, researchers have long used joint models based on traditional machine learning to jointly learn some closely related natural language processing tasks. Early joint learning methods mostly for entity and relation extraction used structured systems based on feature engineering [29, 30], which required complex feature engineering, strongly relied on natural language processing tools, and still led to the problem of error propagation. In 2016, the end-to-end model proposed by Miwa and Bansal [31] laid the foundation for various efficient neural network-based joint extraction models in recent years, but they used a NN structure to predict entity labels, thus ignoring entities long-distance dependencies between tags. Zheng et al. [32] performed joint learning by sharing the underlying expressions of neural networks. Li et al. [33] applied the same method to the extraction of entities and relation in biomedical texts, but the parameter sharing method still has two subtasks, only that there is interaction between these two subtasks through parameter sharing. The training process is still to identify entities firstly and then perform pair-wise matching based on their prediction information to classify relationships. This kind of redundant information will still be generated for entities with no relationship. Zheng et al. [2] proposed a new labelling strategy in 2017. The new labelling strategy turns the relation extraction involving sequence labelling tasks and classification tasks into sequence labelling tasks and uses a end-to-end neural network model to directly obtain entity-relation triples. Our work focuses on the improvement of this model having the architecture shown in Fig. 1, which mainly includes the layers of inputting, embedding, encoding, decoding and outputting.

3 Proposed method

The LSTM-LSTM-Bias joint extraction model uses a static word vector representation for word embedding, which does not take into account that the same word may have different semantics. In this paper, on the basis of the LSTM-LSTM-Bias joint extraction model proposed by Zheng et al. [2], the BERT pre-training model is introduced to realize the modeling of polysemous words, and a joint extraction model BERT-BILSTM-LSTM is proposed.

3.1 Label mode

The BERT-BILSTM-LSTM model adopts the label mode consistent with the LSTM-LSTM-Bias model. This mode is composed of three parts: the location information, the relation type information and the role information of the entities. The B, I, E in the labels represent the starting words, internal words, and ending words of the entities, and S represents the entities that contain only one word. The numbers 1 and 2 in the label indicate the order in which the entities appear in the relationship, where the number 1 indicates the entities that appear first in the relation, and the number 2 indicates the entities that appear later in the relation. For example, the starting word of the entity that appears first in the Country-President relationship can be expressed as “B-CP-1”. In addition, all other irrelevant words are marked as “O”.

3.2 Model structure

The BERT-BILSTM-LSTM model contains a BERT layer, an encoding layer, a decoding layer and a softmax layer. The structure of the model is shown in Fig. 2.

3.2.1 BERT layer

The BERT layer accurately learns the semantic information of words through two steps of pre-training and fine-tuning. First it uses other large corpus to pre-train the BERT model and then solves the joint extraction problem through fine-tuning. We use the access method shown in Fig. 3 to add the BERT model to the joint extraction model. In Fig. 3, E represents the input embedding, Ti is the contextual representation of the word i, and [CLS] is a special symbol for classification output. [CLS] is ignored during joint extraction and marked as “O”. When a sentence of length n is input into BERT, a “[CLS]” symbol is added to the beginning of the sentence, the sentence length becomes $n + 1$, and the corresponding output label adds a label “O”, and the length becomes $n + 1$.

3.2.2 Encoding layer

The BERT layer is followed by the encoding layer, which can learn the representation characteristics of the input data. The encoding layer is a bidirectional LSTM, which consists of two LSTM layers in parallel with a forward LSTM and a backward LSTM. Each LSTM layer is composed of a series of cyclically connected subnets, and each time step is an LSTM memory block. The LSTM memory block calculates the state vector of the hidden layer at the current moment based on the state of the hidden layer at the

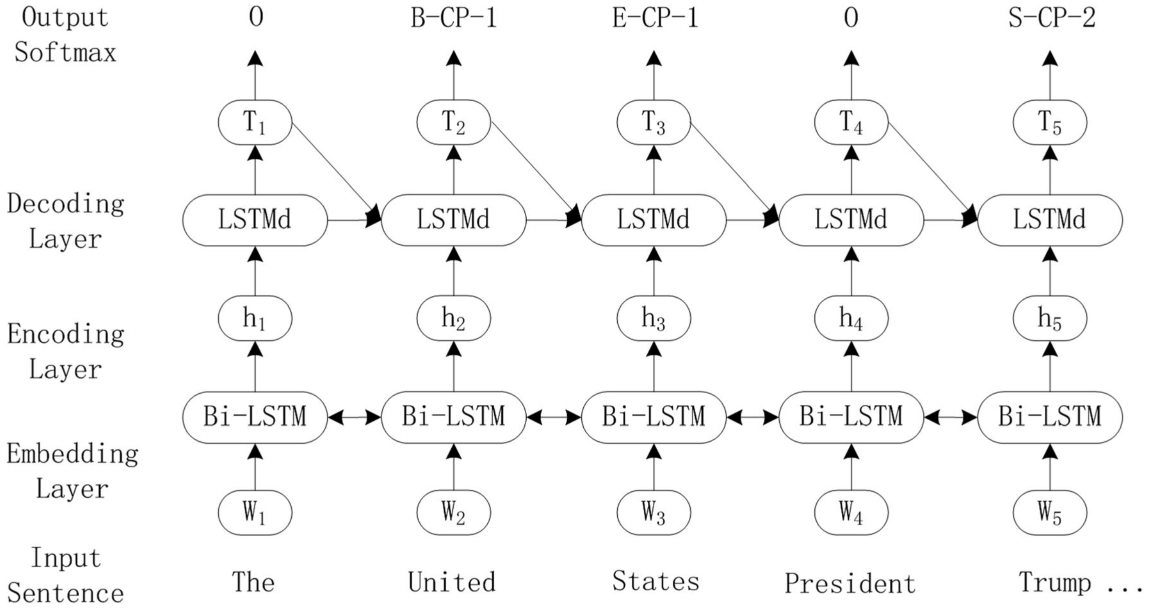


Figure1 End-to-end model proposed by Zheng et al.

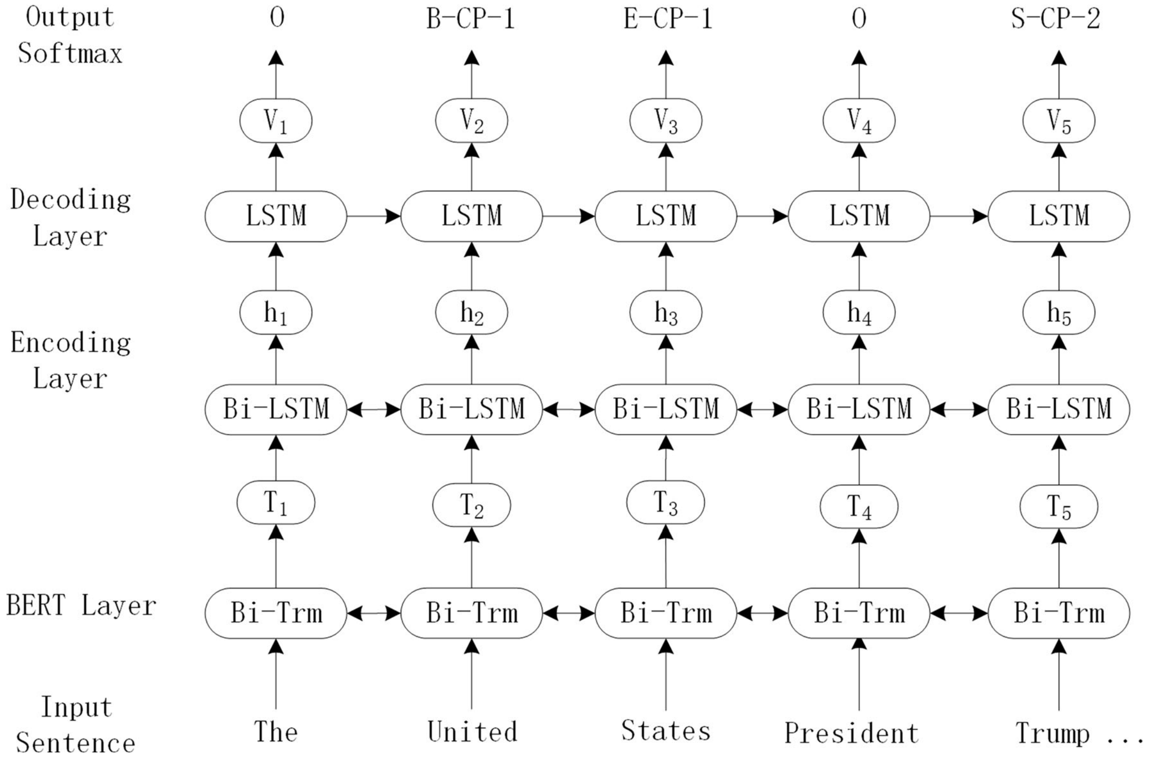


Fig. 2 BERT-BILSTM-LSTM model

previous moment and the output vector of the BERT layer at the current moment. The structure of each LSTM cell is shown in Fig. 4.

The specific calculation formula is as follows:

$$i^{(t)} = \sigma\left(W_{ix}x^{(t)} + W_{ih}h^{(t-1)} + b_i\right) \quad (1)$$

$$f^{(t)} = \sigma\left(W_{fx}x^{(t)} + W_{fh}h^{(t-1)} + b_f\right) \quad (2)$$

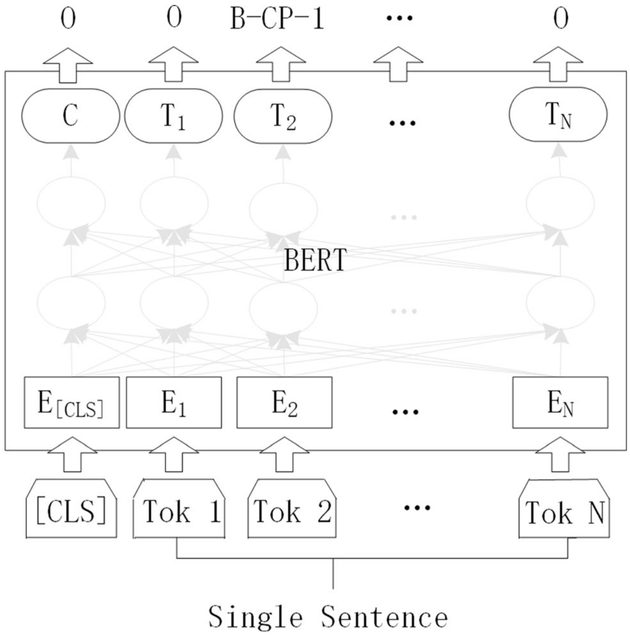


Fig. 3 BERT model is combined with the joint extraction task

$$g^{(t)} = \tanh\left(W_{gx}x^{(t)} + W_{gh}h^{(t-1)} + b_g\right) \tag{3}$$

$$c^{(t)} = i^{(t)} \cdot g^{(t)} + f^{(t)} \cdot c^{(t-1)} \tag{4}$$

$$o^{(t)} = \sigma\left(W_{ox}x^{(t)} + W_{oh}h^{(t-1)} + b_o\right) \tag{5}$$

$$h^{(t)} = \tanh\left(c^{(t)}\right) \cdot o^{(t)} \tag{6}$$

Among them, the formula (1) is the calculation formula of the input gate i, $x^{(t)}$ represents the data of input gate and the current time step t, W_{ix} represents the weight matrix from the BERT layer to the input gate, W_{ih} represents the weight matrix from the hidden state to the input gate, and b_i is the bias term of the input gate. The formula (2) is the calculation formula of the forget gate f, W_{fx} represents the weight matrix from the BERT layer to the forget gate, W_{fh} represents the weight matrix from the hidden state to the forget gate, and b_f is the bias term of the forget gate. c is the cell memory. o is the output gate. The formula (6) is the calculation formula for the output value of the memory cell, and $h^{(t)}$ is the product of the cell memory $c^{(t)}$ and the output gate $o^{(t)}$.

3.2.3 Decoding layer

The encoding layer is followed by the decoding layer, which consists of a single-layer LSTM network, and the function of the decoding layer is to generate tag sequences. The decoding layer uses the output vector $c_2^{(t-1)}$ of the memory unit at the previous moment, the hidden layer state $v^{(t-1)}$ at the previous moment, and the current hidden layer state $h^{(t)}$ of the encoding layer to calculate the hidden layer state value $v^{(t)}$ at the current moment. The specific calculation process is similar to the encoding layer.

3.2.4 Softmax layer

The decoding layer is followed by a softmax layer, which is mainly used for normalization processing. The specific formula is as follows:

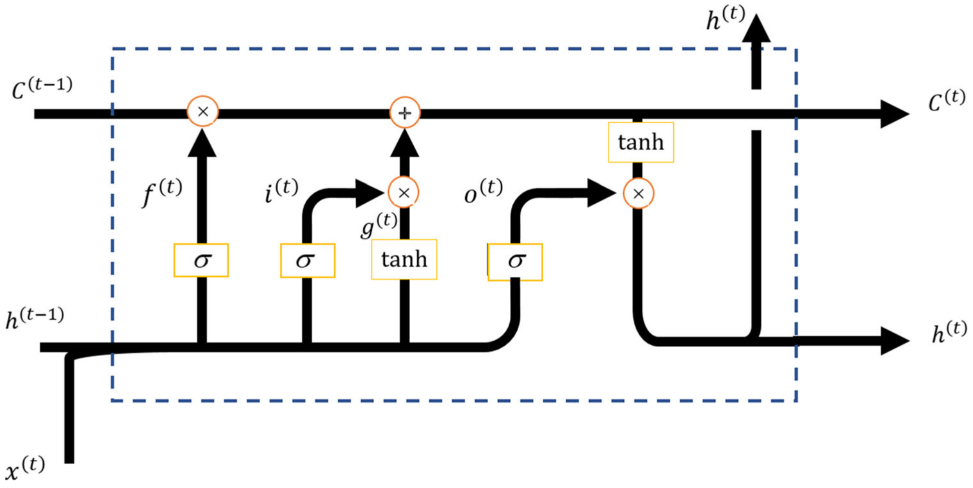
$$y_t = W_y T_t + b_y \tag{7}$$

$$p_t^i = \frac{\exp(y_t^i)}{\sum_{j=1}^{N_t} \exp(y_t^j)} \tag{8}$$

Among them, W_y is the softmax matrix and N_t is the number of marks. At the same time, the objective function L without bias is used. The formula is defined as follows:

$$L = \max \sum_{j=1}^{|D|} \sum_{t=1}^{L_j} \left(\log(p_t^{(j)} = y_t^{(j)} | x_j, \theta\right) \tag{9}$$

Fig. 4 A LSTM cell



|D| is the size of the training set, L_j is the length of sentence x_j, $y_t^{(j)}$ is the true label of the tth word of sentence x_j, and $p_t^{(j)}$ is the normalized probability value of the obtained predicted label.

3.3 Training algorithm

3.3.1 Pre-training

The pre-training of the BERT model needs to use a large corpus, which has high requirements on the performance of the corpus and the server. In this paper, we use the BERT pre-training model disclosed by Google, which includes BERT-Base and BERT-large, and each model has two versions, Uncased and Cased. Among them, the Cased version retains the case of the original text, and the Uncased version converts all uppercase letters in the text to lowercase before word segmentation and removes all spoken marks. Because the tasks in this paper do not require high case sensitivity, the Uncased model is adopted. The download address of all pre-trained models is: https://github.com/google-research/bert.

3.3.2 Network structure setting

Number of LSTM layers (num_layers): The number of LSTMs in the hidden layer.
The size of the state of the LSTM unit(state_size): The size of the state vector of each LSTM memory unit. At each moment, the size of the state vector of the entire hidden layer is state_size*num_layers.
Dimension of LSTM unit output (output_size): The size of the LSTM output unit, which is generally the same as the unit state vector.
Dimension of LSTM unit input (input_size): The size of the LSTM input unit, which is generally the same as the unit state vector.

3.3.3 Model training setting

trains The data used to train the model.
tests: The data used to test the model.
max_seq_length: Sentence truncated length.
vocab.txt: The dictionary used during BERT model training.
bert_config.json: The configuration file of the parameters of the BERT model.
warmup_proportion: The proportion of warm up steps.
learning_rate: The magnitude of the progress in the direction of the gradient.
batch_size: The number of truncated sequences of loss summary. It only updates the gradient after obtaining the loss sum of a batch of sequences.
epoch: The number of times that all training samples repeatedly perform a forward pass and a reverse pass.

3.3.4 Training process

The model training process is shown in Algorithm 1. By modeling polysemous words, the BERT-BILSTM-LSTM joint extraction model can learn different semantic information of the same word according to context information.

Algorithm 1 The training process of BERT-BILSTM-LSTM model

```
Begin
Define the epoch
Define the batch_size
  Define the network structure parameter of BERT,BILSTM and LSTM
  Calculate the num_train_steps, num_train_steps= len(train_examples)//batch_size*epoch
  Calculate the num_warmup_steps, num_warmup_steps = int (num_train_steps*warmup_proportion)
  Replace the training set and test seta data using the BERT word vector pre-trained by Google
  For step in range(num_train_steps):
        （1）Forward propagation of the BERT layer:
        The BERT layer performs forward propagation from start to end in the forward time direction
        （2）Forward propagation of the encoding layer:
        Forward LSTM propagates forward from start to end in the forward time direction
        Backward LSTM propagates forward from start to end in the negative time direction
        （3）Forward propagation of the decoding layer:
        Forward LSTM propagates forward from start to end in the forward time direction
        （4）Decoding layer back propagation:
        Forward LSTM back-propagates from start to end in the negative time direction
        （5）Encoding layer backpropagation:
        Forward LSTM back-propagates from start to end in the negative time direction
        Backward LSTM back-propagates from start to end in the forward time direction
        （6）BERT layer backpropagation for fine-tuning
End
```

4 Experiments

4.1 Experimental environment

The paper carried out experiments on the standard data set NYT and the self-constructed agricultural data set AgriRelation. The server used in the experiments had an Intel Xeon E5-2620 v4 processor and 16G of memory. The experiments were performed on the Ubuntu16.04 operating system, using python3.5 and tensorflow1.10 to build the extraction model and a GPU card of K80 to accelerate the training.

4.2 Data sets

4.2.1 AgriRelation

Since there is no public agricultural relation extraction data set, we constructed the agricultural data set AgriRelation through web crawler technology crawled from Baidu Baike refer to the *Agricultural Thesaurus* [34]. In order to reduce the impact of sparse sample, we choose “fruit” and “geographical location” as entities, and “place of origin” as the entity relation after analyzing the agricultural data in the *Agricultural Thesaurus* and Baidu Baike. So that more sentences in the crawled text data include two entities and the relation. The specific construction steps of the data set are as follows:

1. Crawl text data for various “fruits”. By analyzing the URL address of Baidu Baike, we can know that the Baidu Baike URL has a fixed prefix format: “https://baike.baidu.com/item/term”. Therefore, by replacing the “term” in the URL, you can get the set of seed URLs that need to be crawled. In order to increase the number of positive samples, we select all fruit thesauruses and their aliases under the category of “fruit crops” in the *Agricultural Thesaurus* for crawling.

2. Filter text data that contain “geographic location”. Select all thesauruses of the geographic and administrative districts under the category of “China” in the *Agricultural Thesaurus*, and then parse the text part of the div block with class value of para in the pages of fruit crops obtained in the previous step to extract the sentences containing China’s geographical and administrative districts. At the same time, in order to increase the number of positive samples, we extracted sentences containing words such as “origin” and “producing area”.
3. Process the data and complete the triples. By manually complementing sentences that do not contain complete triples, we get the data set AgriRelation for relation extraction. The AgriRelation contains two parts: training set and test set. The training set contains 1348 sentences and the test set contains 187 sentences.
4. Annotate data. Manual data annotation is performed on the obtained data set. In this paper, we use entity location information, relation type information, and entity role information to label the entities in the triples. For example, the sentence “Baishui County is recognized by experts at home and abroad as one of the best producing areas for apples”, which contains the two entities “Baishui County” and “Apple” and their “producing area” relationship. Baishui County is the first entity, so it is labelled “E1”, and “Apple” is the second entity, so it is labelled “E2”. The “Baishui” in “Baishui County” is the start position of the entity, “County” is the end position of the entity, so they are marked as “E1B” and “E1L” respectively. In the same way, “Apple” is marked as “E2S”.

4.2.2 NYT

In order to be consistent with the experiment of the LSTM-LSTM-Bias model proposed by Zheng et al. [2], we use the NYT public data set to verify the experimental data. The download address of NYT data set is: https://github.com/INK-USC/DS-RelationExtraction. The data set has 24 types of relationships, including two sets of training set and test set. There are 235,982 sentences in the training set and 395 sentences in the test set. Each sentence in the training set consists of 4 parts: “sentText”, “articleId”, “relationMentions”, and “entityMentions”:

- “sentText”: “But that spasm of irritation by …”
- “articleId”: “/m/vinci8/data1/riedel/projects/relation/kb/nyt1/docstore/nyt-2005–2006.backup/1677367.xml.pb”
- “relationMentions”: [{“em1Text”:“Bobby Fischer”,“em2Text”:“Iceland”, “label”:“/people/person/nationality”},……]
- “entityMentions”: [{“start”: 0, “label”:“PERSON”, “text”:“Bobby Fischer”}, ……]

Among them, sentText is the original sentence, articleId is the source of the sentence, and relationMentions is the description of all entity relationships in the sentences. In relationMentions, em1Text represents entity 1, em2Text represents entity 2, label represents the relationship category, and entityMentions is a description of all entities in the sentence. The start in entityMentions represents the entity position number, label represents the entity category, and text represents the entity content.

In order to ensure quality, the test set is manually annotated. The test set contains 24 relation types and 47 entity types. In order to facilitate the comparison of results, we downloaded the data set labelled by Zheng et al. [2] for model training. Since the statements at the end of the training set contain few relationships and most of the corresponding output tags are “O”, we intercept the previous 66,339 sentences as the training set, and the intercepted training set has 162 tags (including the label “O”). In order to access the BERT pre-training model, in addition to the original 162 tags, we added “X”, “[CLS]” and “[SEP]”, resulting in a total of 165 tags. In order to avoid the problem of disappearing gradient when the sentence is too long, we refer to the experiments of Zheng et al. [2] and set the maximum sentence length. When the sentence length exceeds 50, only the first 50 words are kept as sentence input.

4.3 Parameter settings

The experiments use the BPTT algorithm to update the parameters of the model, and use AdamWeightDecayOptimizer to optimize. The num_layers of the encoding layer is 300, the num_layers of the decoding layer is 600, the learning_rate is 5e-5, the batch_size is 64, the warmup_proportion is 0.1, and the sentence truncation length is 50. The epoch on the agricultural data set is 300, and the epoch on the NYT data set is 50. This paper uses the public word vectors set trained from Baidu Encyclopedia Corpus by SGNS to represent the Chinese sentences. The download address is: https://github.com/Embedding/Chinese-Word-Vectors. The size of Chinese word vectors is 300 dimensions.

4.4 Evaluation indictor

In order to evaluate the effect of relation extraction, as mentioned in other documents, we use precision, recall, and F1 to evaluate the experimental results. The formulas are defined as follows:

Table 1 Results of agricultural NER

Elements	Entity1			Entity2		
PRF	Precision	Recall	$F1$	Precision	Recall	$F1$
LSTM-CRF	23.5	12.5	16.3	25	15.6	19.2
LSTM-LSTM	29.4	15.6	20.4	33.3	21.9	26.4
LSTM-LSTM-Bias	35.7	31.2	33.3	36.7	34.3	35.5
BERT-BILSTM-LSTM	46.2	43	44.5	44.6	49.2	46.8
BERT-BILSTM-LSTM-Bias (bias = 10)	44.2	43	43.6	51.2	54.7	52.9

Table 2 Results of agricultural RE

PRF	Precision	Recall	$F1$
LSTM-CRF	83.3	31.2	45.4
LSTM-LSTM	71.4	31.3	43.5
LSTM-LSTM-Bias	61.9	40.6	49
BERT-BILSTM-LSTM	61.5	54.2	57.6
BERT-BILSTM-LSTM-Bias (bias = 10)	60.6	55.5	57.9

Table 4 Results of RE

PRF	Precision	Recall	$F1$
LSTM-CRF	72.4	34.1	46.5
LSTM-LSTM	70.5	34.0	45.8
LSTM-LSTM-Bias	64.5	43.7	52
BERT-BILSTM-LSTM	61.7	51.2	55.9
BERT-BILSTM-LSTM-Bias (bias = 10)	61	51.3	55.7

Table 3 Results of NER

Elements	Entity1			Entity2		
PRF	Precision	Recall	$F1$	Precision	Recall	$F1$
LSTM-CRF	59.6	32.5	42	60.5	32.5	42.3
LSTM-LSTM	59.3	34.2	43.4	61.9	33.4	43.4
LSTM-LSTM-Bias	59.0	47.9	52.9	59.7	45.1	51.4
BERT-BILSTM-LSTM	58.1	55.7	56.9	57.7	54.4	56
BERT-BILSTM-LSTM-Bias (bias = 10)	57.9	56.4	57.1	57.5	55	56.2

$$\text{Precision} = \frac{E_{\text{correct}}}{E_{\text{recognition}}} \tag{10}$$

$$\text{Recall} = \frac{E_{\text{correct}}}{E_{\text{sample}}} \tag{11}$$

$$F1 = \frac{2 * \text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}} \tag{12}$$

Because the BERT-BILSTM-LSTM joint extraction model is not trained with the label of the entity type, there is no need to consider the entity type in the evaluation. When the relation type of the triple and the head offset of the two corresponding entities are correct, the triple is considered correct. E_{correct} represents the number of correct triples identified in the output sequence of the model, $E_{\text{recognition}}$ represents the number of all triples identified in the output sequence of the model, and E_{sample} represents the number of triples contained in the data set. Precision reflects the precision rate, which indicates how many triples identified are correct triples. Recall reflects the recall rate, which indicates how many correct triples have been identified. $F1$ is a comprehensive evaluation of the results of Precision and Recall.

4.5 Results

4.5.1 The experimental results using AgriRelation

In the experiments, we used the evaluation function evaluate_triple in the exaluate.py file written by Zheng et al. [2], which directly returns the evaluation results of entity1, entity2 and relation. In order to make the results objectively, we train the model 5 times to get the prediction results and take the average. The experimental results of all models using the agricultural data set AgriRelation are shown in Tables 1 and 2. It can be seen from the tables that the BERT-BILSTM-LSTM model has obtained the highest F1 value and Recall value both in entity recognition and relation extraction. Experimental results show that the BERT-BILSTM-LSTM model can extract relation effectively when agricultural data sets are in a small corpus. Furthermore, we did another experiment to add a bias loss function to BERT-LSTM-LSTM model, which enhances the relationship between related entity pairs and reduces the influence of invalid entity tags. The experimental results show that the F1 value of BERT-BILSTM-LSTM-Bias is not much better than BERT-BILSTM-LSTM model.

4.5.2 The experimental results using NYT

In order to verify the effectiveness of the BERT-BILSTM-LSTM model, we also conducted experiments using the standard data set NYT. The experimental results of all models on the standard data set NYT are shown in Tables 3 and 4. The results show that the $F1$ value of the BERT-BILSTM-LSTM model is increased by 3.9 percentage points compared with the best results of other models for the NYT standard data set, indicating that the BERT-BILSTM-LSTM model can effectively improve the effect of relation extraction by using the standard data set. Moreover, the Recall has also been significantly improved in relation extraction, that is to say the model can identify more entity relation triples. In addition, we also test the bias model of BERT-BILSTM-LSTM using NYT data set. The experimental results showed that the $F1$ value of BERT-BILSTM-LSTM-Bias model was close to that of the BERT-BILSTM-LSTM model.

5 Conclusion

In this paper, we have improved the LSTM-LSTM-Bias joint extraction model, and proposed a joint model for agricultural entity and relation extraction based on BERT model. By using the characteristics of BERT, that different meanings of the same word can be learned according to the context information. In the experiments, we used the BERT model to replace the commonly used Word2vec model and realized the modelling of polysemous words through pre-training and fine-tuning. It can be seen from Tables 2 and 4 that the $F1$ value of BERT-BILSTM-LSTM model is improved compared with LSTM-LSTM-Bias for the two data sets, which indicates that BERT-BILSTM-LSTM model is an effective relationship extraction model. However, the Recall in Tables 2 and 4 increases while the Precision decreases, indicating that although the model recognizes more entity relations, some entity relations are wrong. As can be seen from Tables 1 and 3, the $F1$ value of proposed model for entity recognition is also improved. On NYT data set, entity recognition results also have the situation that the Recall increases while the Precision decreases. But on the data set AgriRelation, the Precision and Recall of entity recognition are both improved, which indicating that the model is also applicable to small sample data sets. We also compared the experimental results with those of the BERT-BILSTM-LSTM-Bias model. The experimental results show that adding bias function to the BERT-BILSTM-LSTM model will not significantly improve the extraction efficiency.

Acknowledgements Our works have been achieved significant help and supporting from Natural Science Foundation of Hunan Province of China (Grant No. 2019JJ40133), Natural Science Foundation of Hunan Province of China (Grant No. 2019JJ50239), Scientific Research Fund of Hunan Provincial Education Department of China (Grant No. 20A249), as well as the Key Research and Development Program of Hunan Province of China (Grant No. 2020NK2033).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Goldberg Y, Levy O (2014) word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. arXiv:1402.3722
2. Zheng S, Wang F, Bao H, Hao Y, Zhou P, Xu B (2017) Joint extraction of entities and relations based on a novel tagging scheme. arXiv:1706.05075
3. Zhou Z, Shin J, Zhang L, Gurudu S, Gotway M, Liang J (2017) Fine-tuning convolutional neural networks for biomedical image analysis: actively and incrementally. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7340–7351
4. Devlin J, Chang M-W, Lee K, Toutanova K (2018) Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805
5. Bikel DM, Schwartz R, Weischedel RM (1999) An algorithm that learns what's in a name. Mach Learn 34(1–3):211–231
6. Fu G, Luke K-K (2005) Chinese named entity recognition using lexicalized hmms. ACM SIGKDD Explor Newsl 7(1):19–25
7. Chieu HL, Ng HT (2002) Named entity recognition: a maximum entropy approach using global information. In: COLING 2002: the 19th international conference on computational linguistics
8. Uchimoto K, Ma Q, Murata M, Ozaku H, Isahara H (2000),"amed entity extraction based on a maximum entropy model and transformation rules. In: Proceedings of the 38th annual meeting of the association for computational linguistics, pp 326–335
9. Isozaki H, Kazawa H (2002) Efficient support vector classifiers for named entity recognition. In: COLING 2002: the 19th international conference on computational linguistics
10. Chiu JP, Nichols E (2015) Named entity recognition with bidirectional lstm-cnns. arXiv:1511.08308
11. Huang Z, Xu W, Yu K (2015) Bidirectional LSTM-CRF models for sequence tagging. arXiv:1508.01991
12. Ma X, Hovy E (2016) End-to-end sequence labeling via bi-directional LSTM-CNNS-CRF. arXiv:1603.01354
13. Lample G, Ballesteros M, Subramanian S, Kawakami K, Dyer C (2016) Neural architectures for named entity recognition. arXiv:1603.01360
14. Wu H, Lu L, Yu B (2019) Chinese named entity recognition based on transfer learning and bilstm-crf. Small Micro Comput Syst 40:1142–1147

15. Mintz M, Bills S, Snow R, Jurafsky D (2009) Distant supervision for relation extraction without labeled data. In: Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP: volume 2-volume 2. Association for Computational Linguistics, pp 1003–1011
16. Zelenko D, Aone C, Richardella A (2003) Kernel methods for relation extraction. J Mach Learn Res 3(Feb):1083–1106
17. Zhou G, Zhang M, Ji D, Zhu Q (2007) Tree kernel-based relation extraction with context-sensitive structured parse tree information. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 728–736
18. Yao L, Riedel S, McCallum A (2010) Collective cross-document relation extraction without labelled data. In: Proceedings of the 2010 conference on empirical methods in natural language processing. Association for Computational Linguistics, pp 1013–1023
19. Zeng D, Liu K, Lai S, Zhou G, Zhao J (2014) Relation classification via convolutional deep neural network. In: Proceedings of COLING 2014, the 25th international conference on computational linguistics: technical papers, pp 2335–2344
20. Nguyen TH, Grishman R (2015) Relation extraction: perspective from convolutional neural networks. In: Proceedings of the 1st workshop on vector space modeling for natural language processing, pp 39–48
21. dos Santos CN, Xiang B, Zhou B (2015) Classifying relations by ranking with convolutional neural networks. arXiv:1504.06580
22. Socher R, Huval B, Manning CD, Ng AY (2012) 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, pp 1201–1211
23. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H, Xu B (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics (volume 2: short papers), pp 207–212
24. Lin C, Miller T, Dligach D, Amiri H, Bethard S, Savova G (2018) Self-training improves recurrent neural networks performance for temporal relation extraction. In: Proceedings of the ninth international workshop on health text mining and information analysis, pp 165–176
25. Zhang Y, Qi P, Manning CD (2018) Graph convolution over pruned dependency trees improves relation extraction. arXiv:1809.10185
26. Zhu H, Lin Y, Liu Z, Fu J, Chua T-S, Sun M (2019) Graph neural networks with generated parameters for relation extraction. arXiv:1902.00756
27. Shi P, Lin J (2019) Simple bert models for relation extraction and semantic role labeling. arXiv:1904.05255
28. Shen T, Wang D, Feng S, Zhang Y (2019) Bert-based denoising and reconstructing data of distant supervision for relation extraction. In: CCKS2019-shared task
29. Li Q, Ji H (2014) Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd annual meeting of the association for computational linguistics (volume 1: long papers), vol 1, pp 402–412
30. Miwa M, Sasaki Y (2014) Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1858–1869
31. Miwa M, Bansal M (2016) End-to-end relation extraction using lstms on sequences and tree structures. arXiv:1601.00770
32. Zheng S, Hao Y, Lu D, Bao H, Xu J, Hao H, Xu B (2017) Joint entity and relation extraction based on a hybrid neural network. Neurocomputing 257:59–66
33. Li F, Zhang M, Fu G, Ji D (2017) A neural joint model for entity and relation extraction from biomedical text. BMC Bioinform 18(1):198
34. Fang LM et al (1994) Agricultural thesaurus (the third volume). China Agriculture Press, Beijing, pp 191–192