Journal of Computers Vol. 29 No. 6, 2018, pp. 79-89
doi:10.3966/199115992018122906007

Entity Relationship Extraction of Chinese Enterprises on Web Data

Xiaotao Wei*, Yang Liu, Lei Meng, Yu Zhu, Yinglong Li

School of Software Engineering, Beijing Jiaotong University, Beijing, China
{weixt, 17126181, 15121689, 15301057, 17121710}@bjtu.edu.cn

Received 1 June 2018; Revised 1 July 2018; Accepted 1 August 2018

Abstract. Enterprise entity relationship extraction is an important part of entity relationship extraction. Extracting corporate relationships from open data is of great significance in market analysis and selection of business partners. Due to the complexity of grammar and flexible expression in Chinese language, the traditional method for extracting Chinese enterprise entity relationship has a very poor effect. We propose an algorithm based on the integration of dependency grammar analysis of self-adaptive attention mechanism and long short-term memory network (DEP_ATT_LSTM) by vectorizing the text on which word segmentation is performed and inputting it into the LSTM network to obtain the text feature representation of sentences, then adopting self-adaptive attention mechanism based on dependency parser to calculate the weight of the text feature, and sending the obtained feature vectors into a classifier for entity relationship extraction. Experiments prove that the algorithm performs well. The accuracy, recall rate and F1 value reach 83.23%, 89.55% and 86.81%, respectively.

Keywords: attention mechanism, dependency parser, enterprise entity relationship extraction, LSTM

1 Introduction

Massive information has penetrated into social economy in the form of text. How to find useful information from these structured or semi-structured data has become a hot topic in recent years.

Entity relationship extraction is an important part of Natural Language Processing (NLP) [1], and it is a typical information extraction problem. In financial field, enterprise entity relationship mining technology research is based on enterprise entity identification [2]. Accurately identifying enterprise entity information and relationships from open data are of vital importance for enterprises. At present, most of the enterprise entity relationship extraction systems are based on industrial and commercial data. Their search scope is limited. However, as informatization degree deepens, enterprise relationships may be implicit in various data carriers. For example, media reports may involve the relationship of cooperation and so on. Bulletins and reviews may also include relevant information.

Therefore, we crawl the news from the financial websites in China, and conduct research on the extraction of Chinese enterprise entity relationships.

2 Related Works

The extraction of English enterprise entity relationships has recently achieved breakthrough progress, while the relevant research on Chinese develops slowly due to the complicated Chinese grammar, flexible expression, and lacking corpus, etc.

At present, the methods of entity relationship extraction mainly include the entity relationship extraction basded on pattern matching, the entity relationship extraction basded on machine learning and the entity relationship extraction basded on deep learning. However, people tend to use neural networks

* Corresponding Author

to solve problems and automatically extract relevant features from data samples.

When using these methods to extract entity relations, we are faced with two challenges: the first is the extraction of labeled relations between two entities; the second is that the same pair of entities may have multiple labels and it is unclear which label dominates. In order to slove this problem, Surdeanu et al. [3] put forward a new approach to multi-instance multi-label learning for relation extraction, which jointly models all the instances of a pair of entities in text and all their labels using a graphical model with latent variables. Long Short-term Memory (LSTM) is an improvement of Recurrent Neural Networks (RNN). By constructing memory units to store historical information, the long-distance dependence of RNN and CNN is effectively alleviated. Zeng et al. [4] propose Piecewise Convolutional Neural Networks (PCNNs) with multi-instance learning for remotely supervised relation extraction, and the uncertainty of instance labels is fully taken into consideration. In addition, they adopt a convolutional architecture with piecewise max pooling to automatically learn relevant features, which reduces the noise aroused by the feature extraction process. Attention Mechanism is a model simulating the mechanism of human brain attention. By calculating the distribution probability of attention, some key inputs are highlighted and the influence on the output characteristics is increased. It has a good optimization effect on the neural network model. Lin et al. [5] propose a sentence-level attention-based model for relation extraction which adopts convolutional neural networks to embed the semantics of sentences. Afterwards, they build sentence-level attention over multiple instances, which is expected to dynamically reduce the weights of those noisy instances. Dependency Parser (DP) is a syntactic analysis of sentences after analyzing the dependencies between the components of the sentence. Gan Lixin et al. [6] propose a novel method based on syntactic and semantic features, which can effectively improve the performance of entity relationship detection and extraction on a real text corpus in tourism domain. Etzioni et al. [7] use syntactic analysis to express entity relationships with verbs, and achieve a high precision rate in open entity recognition.

The current enterprise entity relationship extraction mainly draws on the traditional method, and there is little research in open web data. Because of the limitations of Chinese grammar expression and lack of Chinese corpus, it is very difficult to directly apply the method of extracting English entity relationships to the Chinese financial field. Therefore, we propose an attention mechanism integration LSTM algorithm based on dependency parser (DP_ATT_LSTM).

3 Attention Mechanism Integration LSTM Algorithm Based on Dependency Parser

3.1 Chinese Enterprise Entity Relationship Corpus

We focus on enterprise entity relationship extraction of news in Chinese financial area, and use a triple (company entity e_1, company entity e_2, company relationship) to define the entity relationships. By analyzing news text and enterprise relationships in financial area, we make a targeted selection over six types of common enterprise relationships to conduct experiments, all of which are as shown in Table 1.

Table 1. Enterprise relationships and labelling information

Enterprise Relationships	Relationship Description	Relationship Labels
Cooperation	It generally refers that relationships of a series of commercial activities such as capital contribution, cooperation, joint investment, and joint venture exist between enterprises.	Cooperate
Subsidiary	One company is a subsidiary or subordinate company of the other.	Subsidiary
Stock	Company A is the shareholder of company B, or, company A is the holding company of company B.	Stock
Acquisition	An investment behaviour that the enterprise obtains part or all of the ownership of a certain enterprise through certain procedures and means.	Buy
Merger	A behavioural process in which two or more enterprises combine assets to form a new enterprise by concluding and signing a merger agreement in accordance with the relevant laws and regulations.	Merge
Establishment	It generally refers to the subordination relationship between two company entities. The establishment relationship generally reflects the control relationship between the company entities. For example, Company A funds to establish Company B.	Establish

Before constructing the model, we perform a pre-processing on corpus, including data cleaning, segmentation, entity identification, dependency parsing, etc., to obtain data of two or more company entities at the sentence level, and then compose the corpus.

3.2 Attention Mechanism Integration LSTM Method Based on Syntactic Parsing

In multi-classification tasks, the key to accomplishing tasks with high quality is selecting out features with higher discriminability. Lin et al. [8] propose a keyword-based web searching method which uses query document words as the entry point and achieves a high recall rate. Wang et al. [9] propose an extraction method based on a convolutional neural network and keyword strategy. Based on features of the word embeddings, the keyword features are acquired by the term proportion-inverse sentence proportion (TP-ISP) keywords extraction algorithm based on sentences. In addition, there are various relational labels for various entities in the same sentence and the contribution of the same word may not be equal for different interest relationships. So, in extracting relationships, we also need to consider the location of entity pair and contextual semantics. Qin et al. [10] propose an Entity-pair-based Attention Mechanism to highlight the words close to position features, which achieved F1-score of 84.7%.

Although the above methods have achieved certain effects in the extraction of English entity relationships, the effect on text processing in the Chinese financial field is not satisfactory. Therefore, based on LSTM [16], we introduce the attention mechanism based on syntactic parsing, and propose the attention mechanism integration LSTM model based on dependency parsing. The adoption of LSTM avoids the long-distance dependence problem of traditional methods. At the same time, the use of self-adaptive attention mechanism based on syntactic parsing fully considers the relevance of predicate phrases, so we can obtain more effective semantic information. The model mainly comprises the following parts:

3.3 Word-representation Input Level

The simplest method of word representation is one-hot representation method, also known as word bag representation model. It represents each word as an N-dimensional vector in which one row has only one “1” and the rest are “0”. “1” indicates the subscript of the word in the word list, and N indicates the size of the word list. For example: “公司 (Company)” is represented as [0, 0, 0, 1, 0, …, 0], and “公司 (Company)” is subscripted as “4” in the word list. Although the word bag model is simple in representation and has certain robustness, it also has certain defects: first, words are independent with each other, which is easy to cause curse of dimensionality and other problems; second, the method cannot get the meaning of the word itself.

Therefore, we use word embedding to represent the input words. This technology adopts a dense feature vector representation to replace the original one-hot sparse vector for representing the vocabulary [11], which can fully retain semantic and syntactic information of the article. For example, given the sentence S = {2015 年 5 月，公司 A 以 3.90 亿元完成对 B 公司 100%股权收购。(In May 2015, company A completed a 100% equity acquisition to company B by 390 million yuan.)}, the sentence segmentation sequence $S_w = \{w_1, w_2, ..., w_n\}$ is obtained through the word segment tool, that is, S_w = {2015 年，5 月，公司 A，以，3.90 亿，元，完成，对，B 公司，100%，股权，收购 (In 2015, May, company A, by, 390 million, yuan, completed, to, company B, 100%, equity, acquisition)}, wherein n is the number of words after the word segmentation of sentences, and w_i represents the i-th word in S_w. We use a word embedding matrix $W^{emb} \in R^{|d|*|V|}$ to map each word w_i to a word vector W_i, and take it as a feature of the vocabulary level. Wherein $|d|$ is the dimension of the word vector, and $|v|$ is the size of the word list. Thus, a given sentence S can be represented as a matrix $[W_1, W_2, ..., W_n]$ composed of word vectors, and used as an input of the long short-term memory level.

3.4 Long Short-term Memory

In this level, each input will be converted to a feature representation, as shown in Fig. 1. As a special type of RNN, LSTM model can make full use of text sequence information. A LSTM model consists of multiple LSTM units, each of which contains a forgot gate f, an input gate x, an output gate h. Taking the text sequence S_w = {2015 年，5 月，公司 A，以，3.90 亿，元，完成，对，B 公司，100%，股权，

收购 (In 2015, May, company A, by, 390 million, yuan, completed, to, company B, 100%, equity, acquisition)} as an input, and the i-th word as an example to activate the memory unit to obtain the characteristic values of all states of LSTM units:

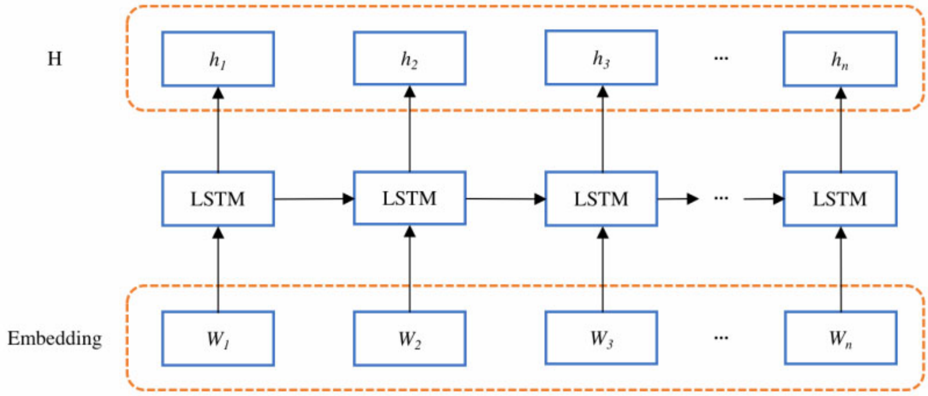

Fig. 1. The architecture of a LSTM

Step 1: deciding what information to drop from the cell, also known as the "forgot gate f_t", which reads h_{t-1} and x_t and outputs a value between 0 and 1 to the C_{t-1} in the current cell state. "0" means "completely discarded" and "1" means "completely retained". See Equation 1.

$$f_t = \sigma\ (W_f \cdot [h_{t-1}, x_t] + b_f). \tag{1}$$

Wherein, h_{t-1} represents the output information of the previous cell, x_t represents the input information of the current cell, σ is a sigmoid function, W_f represents the weight matrix of LSTM, and b_f represents the offset vector of LSTM.

Step 2: determining which part of the new information will enter the cell state. This part is a two-step process: first, the sigmoid layer determines which information needs to be updated, as shown in formula 2; second, the tanh layer will create a new candidate value vector $\tilde{C}$, introducing into the cell state, as shown in Equation 3.

$$i_t = \sigma\ (W_i \cdot [h_{t-1}, x_t] + b_i). \tag{2}$$

$$\tilde{C} = \tanh\ (W_C \cdot [h_{t-1}, x_t] + b_C). \tag{3}$$

Step 3: updating the state of the old cell, that is, updating C_{t-1} to C_t. See Equation 4.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}. \tag{4}$$

Step 4: determining the output of the cell state. First, which part of the cell state will be output is determined by the sigmoid layer, then is processed by the tanh layer and then is multiplied pointwise with the sigmoid layer, and finally, the output part is determined. See Equation 5 and Equation 6.

$$o_t = \sigma\ (W_o \cdot [h_{t-1}, x_t] + b_o). \tag{5}$$

$$h_t = o_t * \tanh\ (C_t). \tag{6}$$

3.5 Attention Mechanism Level Based on Dependency Parsing

Dependency parsing reveals the syntactic structure by analyzing the dependencies between sentence components. In other words, dependency parsing identifies grammatical components such as "subject, predicate, object, attribute, adverbial and complement" in the sentence. It also analyzes the relationships between these components. For example, performing a syntactic analysis on the text "2015 年 5 月，万润科技以 3.90 亿完成对深圳日上光电有限公司 100%股权收购。(In May 2015, Mason Technology completed a 100% equity acquisition to Shenzhen Rishang Optoelectronics Co., Ltd. by 390 million yuan.)", as shown in Fig. 2, the core predicate of the sentence is "完成 (Complete)", the subject is "万润

科技(Mason Technology)", the object is "深圳日上光电有限公司 (Shenzhen Rishang Optoelectronics Co., Ltd.)". The labelling relationships of dependency parsing and their meanings are as shown in Table 2:

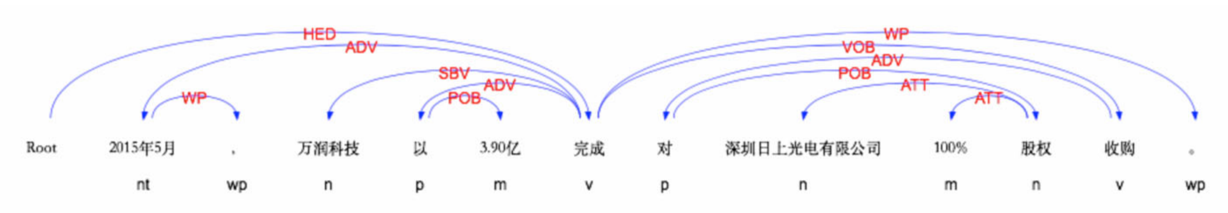

Fig. 2. Dependency parsing diagram

Table 2. Dependency parsing and labelling relationships

Relationship Type	Tag	Description	Example
Subject-predicate relationship	SBV	Subject-verb	I gave him a bunch of flowers (I<-- gave)
Verb-object relationship	VOB	Verb-object	I gave him a bunch of flowers (gave-->flowers)
Indirect-object relationship	IOB	Indirect-object	I gave him a bunch of flowers (gave--> him)
Attribute relationship	ATT	Attribute	Red apple (red<-- apple)
Fronting-object relationship	FOB	Fronting –object	He read all books (books-->read)
Preposition-object relationship	POB	Preposition-object	In the trade zone (in)
Verb-complement relationship	CMP	Complement	Finishing the homework (finishing)

Through observation, Chinese enterprise entity relationship recognition is often closely related to predicates and predicate phrases in sentences. Moreover, predicates in Chinese sentence expressions are often composed of verbs or verb phrases, and exist as important information in the statement. As shown in Fig. 2: the relationship between entity e_1 "万润科技 (Mason Technology)" and entity e_2 "深圳日上光电有限公司 (Shenzhen Rishang Optoelectronics Co., Ltd.)" is "收购 (Acquisition)". Under the premise of dependency parsing, we can observe the dependency arc as verb-object relationship "完成收购 (Complete Acquisition)". We have summarized four syntactic dependence relationships: Verb-object relationship, Indirect-object relationship, Attribute relationship, and Verb-complement relationship. All of them are used as the basis for extracting the predicate and the predicate phrase, and to calculate the probability of the attention mechanism.

In recent years, attention mechanism has been widely used in deep learning. When people read an article, they pay more attention to valuable words so as to understand the main meaning of the article. When dealing with NLP tasks, especially relationship extraction, not every word of word segment is in the same position for task results. Inspired by this, we introduce self-adaptive attention mechanism, which uses the predicates and predicate phrases of sentences to calculate the attention of each segment to extraction results. The model architecture is as shown in Fig. 3. We use Technology Platform of Harbin Institute of Technology (LTP) to analyze sentences syntactically, obtaining the syntactic dependency tree. According to the four kinds of dependence relationships mentioned above, the word sequences $S_{dm} = \{w'_1, w'_2, \ldots, w'_n\}$, namely, S_{dm} = {完成收购 (Complete Acquisition), 股权收购 (Equity Acquisition), … } of predicates and predicate phrases in sentences are extracted, wherein n is the number of predicates or predicate phrases extracted. We perform a same vectorization processing operation on the extracted sequence as before, getting the predicate sequence matrix $S_{dm} = [W'_1, W'_2, \ldots, W'_n]$, and then take the matrix as a LSTM network input, using the last hidden state u_n of the LSTM network output to calculate the weight of attention vector. As shown in Equation 7.

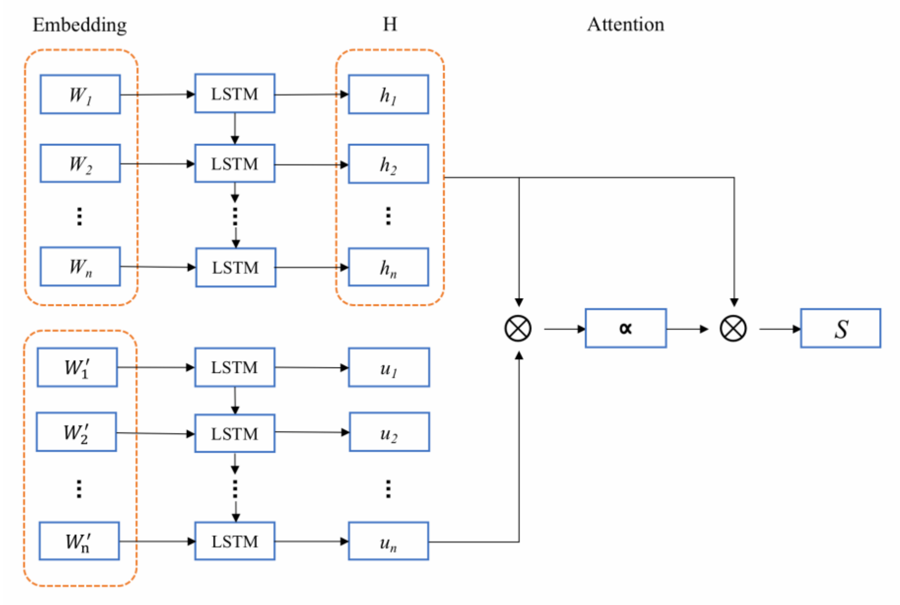

Fig. 3. The architecture of attention mechanism integration LSTM algorithm based on dependency parser

$$u_n = LSTM\,(S_{dm}). \quad \textbf{(7)}$$

$H = [h_1, h_2, h_3, \ldots, h_n]$ is the output matrix of the long short-term memory level. We use the dot product of u_n^T and H to represent the contribution of the word w_i in the S_w to the classification result which is then subject to a standard operation through *softmax* function, obtaining the attention weight of the word w_i to the classification result. The calculation process is shown in Equation 8.

$$\alpha = softmax\,(u^T{}_n H). \quad \textbf{(8)}$$

Finally, we use the obtained attention weight α to calculate the feature vector S, finally extracted from the sentence, as shown in Formula 9. Then we send it to a *softmax* classifier for the extraction of the entity relationship, as shown in Equation 10.

$$S = H \cdot \alpha^T. \quad \textbf{(9)}$$

$$y = softmax\,(W_s S + b_s). \quad \textbf{(10)}$$

4 Experimental Results and Analysis

4.1 Experimental Data

We crawl 1,000 news from authoritative Chinese financial websites (including Sina Finance, Hexun, etc.). Then, we clean the data to extract 888 valid information by using regular expressions, HTML Parsing and other tools. After that, we use NLP tools to process the extracted text and select sentence level data including two or more company entities to conduct experiments. In addition, we label the six types of company relationships. Finally, we use a cross-validation method to verify each other to ensure the accuracy and completeness of the labelled data.

4.2 Evaluation Index

The $\text{Precision} = \frac{\text{out\_right}}{\text{out all}}$, $\text{Recall} = \frac{\text{out\_right}}{\text{tr\_right}}$, and $\text{F1} = \frac{\text{out\_right}}{\text{out all}} = \frac{2 * \text{precision} * \text{recall}}{\text{precison} + \text{recall}}$ are used as the evaluation indexes of the model. Wherein, out_right indicates the number of correct relationships of the output, out_all indicates the number of all the relationships identified, and tr_right indicates the number of all the relationships in the test set.

4.3 Experimental Params

The activation function of this model is ReLU, and the hidden layer has 128 nodes. The *softmax* is used as a classifier. The dropconnect [12] is introduced during training to prevent over-fitting, and the Dropout Rate is 0.5. The number of iterations is 100 and the maximum sequence value is 45.

The standard way to model a neuron's output f as a function of its input x is with $f(x) = tanh(x)$ or $f(x) = sigmoid(x)$. However, in terms of training time with gradient descent, these saturating nonlinearities are much slower than the non-saturating nonlinearity $f(x) = \max(0, x)$. So Krizhevsky et al. [13] propose the ReLU Nonlinearity which can achieve more efficient gradient descent and back propagation avoids the problem of gradient explosion and gradient disappearance.When choosing activation functions, we conduct four sets of comparative experiments: applying *tanh* function, *sigmoid* function, *maxout* function, and ReLU to an attention mechanism integration LSTM algorithm based on dependency parser respectively, as shown in Table 3. The *maxout* and ReLU are much better than the traditional activation function *tanh* and *sigmoid*. Especially DP_ATT_LSTM_ReLU, F1 value is 3.69 points higher than DP_ATT_LSTM_tanh.

Table 3. Results of activation function select

Model	F1 (%)
DP_ATT_LSTM_tanh	83.12
DP_ATT_LSTM_sigmoid	83.25
DP_ATT_LSTM_maxout	84.83
DP_ATT_LSTM_ReLU	86.81

The *tanh* function formula is shown in Equation (11), corresponding to DP_ATT_LSTM_tanh.

$$\tan h(x) = \frac{\sin h(x)}{\cos h(x)}. \tag{11}$$

The *sigmoid* function formula is shown in Equation (12), corresponding to DP_ATT_LSTM_sigmoid.

$$\mathrm{sigmoid(x)} = \frac{1}{1+\exp(-t)}. \tag{12}$$

The *maxout* function formula is shown in Equation (13, 14), corresponding to DP_ATT_LSTM_ maxout. This method was proposed by Goodfellow [14] in the 30th International Conference on Machine Learning, and achieved high effect on MNIST dataset and so on.

$$h_i(x) = \max_{j \in [1,k]} Z_{i\,j}. \tag{13}$$

$$Z_{ij} = x^T W \ldots ij + b_{ij},\ W \in R^{d*m*k}. \tag{14}$$

4.4 Experimental Results

We randomly extract 80% of corpus as the training set and use the remainder as the test set to test the trained model. We adopt SVM [15], kNN [16], and LSTM [17] algorithms to conduct experiments. The effectiveness of keyword strategy on entity relationship extraction is verified. The results are shown in Table 4 and Fig. 4. When the keyword strategy is used to extract entity relationships, the precision rate of three algorithms increases around 2 points. The result indicates that the keyword strategy is conducive to the Chinese entity relationship extraction.

Table. 4. Contrast experiment of keywords strategy

Model	Precision (%)
SVM	60.10
KNN	75.00
LSTM	81.30
SVM_Keywords	62.00
KNN_Keywords	78.00
LSTM_Keywords	82.12

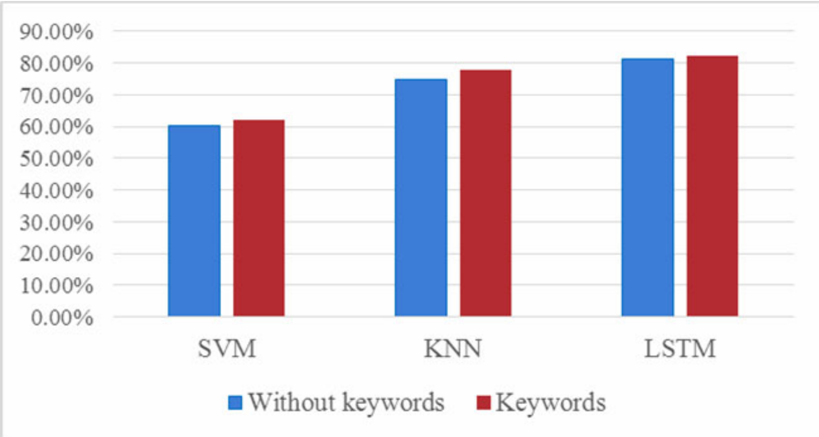

Fig. 4. Contrast experiment of keywords strategy

We select the LSTM algorithm with best performance for the next experiments. Here, we conduct experiments on entity pairs and contextual semantics (Entity_LSTM) and dependency parsing (DP_LSTM), as shown in Table 5. From the experimental results we can find that the precision, recall rate, and F1 values of the model are 81.41%, 80.02% and 81.12% in consideration of entity pairs and contextual semantics. And the expermential indexes are much higher when we use dependency parsing. It can be seen that the dependency parser is more useful than entity pairs in the Chinese entity relationship extraction. So we should fully consider the grammatical structure and semantic logic of the sentence itself apart from the influence of keywords and entity pairs.

Table 5. Contrast of experimental results of entity relationship extraction

Model	Precision (%)	Recall (%)	F1 (%)
LSTM	81.30	79.55	80.42
Entity _LSTM	81.41	80.02	81.12
DP_LSTM	82.00	82.78	82.16
Entity_ATT_LSTM	82.76	83.33	83.00
DP_ATT_LSTM	83.23	89.55	86.81

On this basis, to prove the validity of attention mechanism, we apply it into DP _LSTM method and Entity _LSTM method, as shown in Table 5 and Fig. 5. The experiment shows that the method with attention mechanism has more higher precision, and increased recall rate and F1 value more than 2 percentage points. Furthermore, DP_ATT_LSTM method performs better than the Entity_ATT_LSTM method in terms of the precision rate, recall rate and F1 values, especially in the recall rate. The DP_ATT_LSTM method is more than 6 percentage points higher: 83.33% vs. 89.55%.

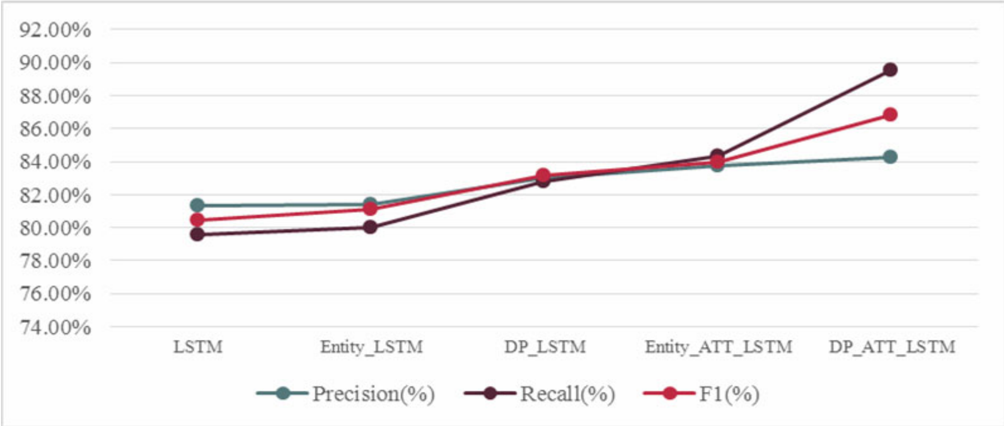

Fig. 5. Contrast of experimental results of relationship classification

Therefore, the model training of Chinese entity relationship extraction should not only take into account part of speech, but also pay attention to the dependency relationship between predicates or predicate phrases and the syntactic structure of the data.

Due to the complexity of the model and the unbalanced distribution of data, there are some over-fitting problems in the experiment. In order to solve this problem, we replace Dropout with Dropconnect [12]. DropConnect is the generalization of Dropout in which each connection, rather than each output unit, can be dropped with probability $1 - p$. DropConnect is similar to Dropout as it introduces dynamic sparsity within the model, but differs in that the sparsity is on the weights W, rather than the output vectors of a layer. The principles of Dropout is shown in Fig. 6 and the principles of DropoutConnect is shown in Fig. 7.

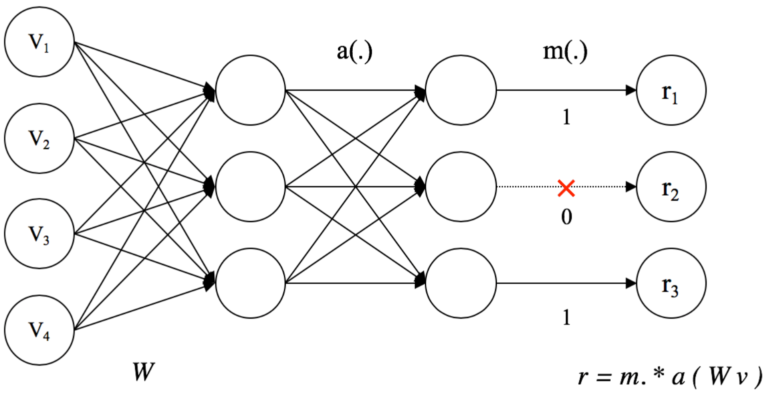

Fig. 6. DropOut Network principle

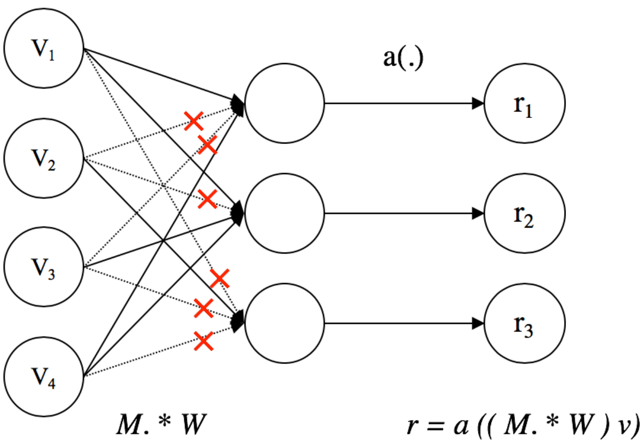

Fig. 7. DropConnect Network principle

From Fig. 6 and Fig. 7, we can find the fully connected layer with DropConnect becomes a sparsely connected layer in which the connections are chosen at randomly during the training stage. DropConnect does not operate on the output of the hidden layer and stochastically chooses the connection between the input data and the nodes to block. The corresponding formula is shown in expressions (15) and (16) as follows. Through the over-fitting experiments of the two methods, it can be seen from Table 6 that DropConnect has a better effect on the model.

$$DropOut: r = m. * a\ (W\ v) \tag{15}$$

$$DropConnect: r = a\ ((M. * W)\ v) \tag{16}$$

Table 6. Results of fight against over-fitting experimental

Model	F1(%)
DP_ATT_LSTM (DropOut)	85.90
DP_ATT_LSTM (DropConnect)	86.81

5 Conclusion

Due to the complexity of Chinese grammar expression and the lack of Chinese corpus, it is very difficult to directly apply the traditional methods to the Chinese financial field. Therefore, we propose an algorithm based on the integration of dependency grammar analysis of self-adaptive attention mechanism and LSTM network. Our main contribution is to calculate the attention weight features through the attention mechanism according to four kinds of predicate dependencies. The experimental results show that the precision rate, recall rate and F1 values are greatly improved compared with other deep learning methods. Besides, the method solves the problems caused by uneven data distribution, and greatly enhances the performance of Chinese entity relationship extraction.

Acknowledgments

This research was funded by the Innovation and Entrepreneurship Project of Beijing Jiaotong University.

Reference

[1] Y. Ke, M. Hagiwara, A natural language processing neural network comprehending English, in: Proc. 2015 International Joint Conference on Neural Networks (IJCNN). IEEE, (2015) 1-7.

[2] L. Meng, Z.-H. Wei, T.-Y. Hu, Q. Cheng, Y. Zhu, X.-T. Wei, An improved method for Chinese company name and abbreviation recognition, in: Proc. KMO 2017: Knowledge Management in Organizations, 2017.

[3] M. Surdeanu, J. Tibshirani, R. Nallapati, D.C. Manning, Multi-instance multi-label learning for relation extraction. in: Proc. EMNLP-CoNLL, 2012.

[4] D. Zeng, K. Liu, Y. Chen, J. Zhao, Distant supervision for relation extraction via piecewise convolutional neural networks, in: Proc. Conference on Empirical Methods in Natural Language Processing, 2015.

[5] Y. Lin, S. Shen, Z. Liu, H. Luan, M. Sun, Neural relation extraction with selective attention over instances, in: Proc. the 54th Annual Meeting of the Association for Computational Linguistics, 2016.

[6] L. Gan, C. Wan, D. Liu, Q. Zhong, T. Jiang, Chinese named entity relation extraction based on syntactic and semantic features, Journal of Computer Research and Development 53(2)(2016) 284-302.

[7] O. Etzioni, A. Fader, J. Christensen, S. Soderland, Mausam, Open information extraction: the second generation, in: Proc. the 22nd International Joint Conference on Artificial Intelligence, 2011.

[8] G.-L. Lin, H. Peng, Q.-L. Ma, J. Wei, J.-W. Qin, A keyword-based method for diversification of web search results, Journal of South China University of Technology (Natural Science Edition) 39(5)(2011) 102-107.

[9] L.-Y. Wang, L. Wang, T.-Y. Zheng, Entity relationship extraction method based on convolutional neural network and keyword strategy, Pattern Recognition and Artificial Intelligence 30(5)(2017) 465-472.

[10] P. Qin, W. Xu, J. Guo, Designing an adaptive attention mechanism for relation classifica-tion, in: Proc. International Joint Conference on Neural Networks, 2017.

[11] R. Collobert, J. Weston, L. Botton, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, The Journal of Machine Learning Research 12(2011) 2493-2537.

[12] L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, R. Fergus, Regularization of neural networks using dropconnect, in: Proc. the 30th International Conference on Machine Learning, 2013.

[13] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, in: S.A. Solla, T.K. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems, MIT Press, Cambridge, 2012, pp. 1097-1105.

[14] I. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout networks, Proceeding of the 30th International Conference on Machine Learning, PMLR 28(3)(2013) 1319-1327.

[15] C. Cortes, V. Vapnik, Support-vector networks, Machine Learning 20(3)(1995) 273-297.

[16] T.M. Cover, P.E. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13(1)(1967) 21-27.

[17] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Computation 9(8)(1997) 1735-1780.