Advances in Computer, Signals and Systems (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/acss.2023.070111
ISSN 2371-8838 Vol. 7 Num. 1

Research on Named Entity Recognition Method Based on Language Pre-Training Model

Jiurong Fan

School of Mechanical and Electrical Engineering, Yunnan Open University, Kunming, China
33768022@qq.com

Keywords: Natural language processing, language pre-training model, transfer learning

Abstract: Aiming at the problem that the existing named entity recognition models have insufficient ability to recognize common unknown words in data, this paper proposes a text vectorization representation method based on language pre-training model. The program can't understand the text directly, and it can only be understood by the program after the text is converted into a numerical value. Firstly, this paper introduces the methods of word vector representation, including discrete representation and distributed representation. The traditional word vector representation method can't deal with the problem of polysemy and can't fully express semantic features. Aiming at the defects of word vector method, this paper proposes a text vectorization method based on language pre-training model. The idea of fine-tune is introduced, and the pre-training model, which completed training on massive data sets, is transferred to the People's Daily data set, and the parameters are optimized. Finally, this paper designs a comparative experiment on the People's Daily data set, compares it with the traditional word embedding methods using CBOW, Skip-gram and GloVe, analyzes the results, and verifies the effectiveness of the proposed method.

1. Introduction

The word embedding method has greatly improved the processing ability of the machine to natural language. However, due to its static nature, the vector through word embedding can only represent one word, and its semantic expression ability is weak, which leads to the processing ability of word embedding method being greatly weakened in the Internet age when the data volume is increasing and all kinds of new words are constantly emerging. Since Devlin[1] and others put forward BERT pre-training model in 2018, and achieved the best results in many natural language processing tasks of that year, the language pre-training model has been widely concerned. This kind of model thinks that for different data sets, the bottom content learned by the model is basically the same, but there are obvious differences in the top content. Therefore, the model can be trained on a large number of general data, and then optimized on specific professional data to achieve the effect of learning new data features. Language pre-training model includes neural network structure and trained parameters. When learning other tasks, it can achieve good results without much data. In this paper, a text vectorization representation method based on language pre-training model is proposed to solve the problem that word vectors can't express polysemous words, and enhance the expression ability of word vectors to text semantics. Thereby improving the recognition ability of the model to the text.

2. Word Vector Representation

The computer can't recognize text characters. To make the computer understand the meaning of text representation, it is necessary to vectorize the text. There are two commonly used expressions of vectors: discrete expression and distributed expression. The discrete representation method expresses the text as a numerical value, which cannot express the semantic similarity between words. Discrete representation represents the text as a vector, which can represent the semantic similarity between words.

2.1. Discrete representation

The discrete representation methods include: one-hot coding, word bag model, word frequency-inverse document frequency method (TF-IDF) model and vector space model (VSM) [2]. One-hot coding method uses a word vector with the same dimension as the dictionary length. The position of the word is set to 1, and the other positions are set to 0. Bag of Words is a model that focuses on words and their frequency of occurrence, and it has nothing to do with word order. TF-IDF is a modification of the bag-of-words method. By adding weight to words, it reflects the importance of words. VSM maps the features of documents to high-dimensional space.

2.2. Distributed representation

Bengioand others put forward a language model based on neural network[3] , and in 2003, they trained word vectors through this model. This kind of vector trained based on language model is called distributed representation of text, which has the ability to express extensive information through smaller dimensions. At the same time, it can better express the relationship between words and the importance of words. For example, in Figure 1, there are two-dimensional vector forms of words such as king, queen, male and female after word embedding: $\overrightarrow{king} + \overrightarrow{male} = \overrightarrow{queen} + \overrightarrow{female}$

As can be seen from the above equation, word embedding reflects the semantic relationship between texts. There are two commonly used distributed representation methods, including word2vec word embedding and glove.

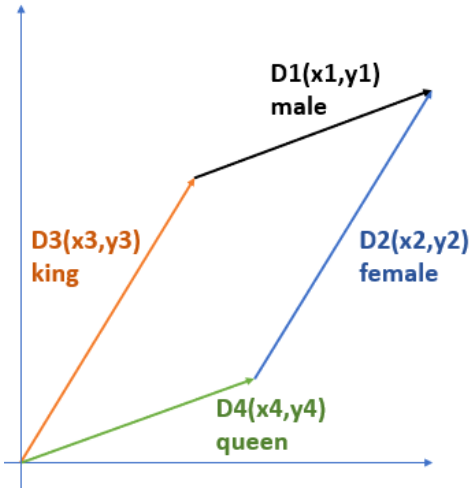

Figure 1: Schematic diagram of distributed text representation

1) Word2Vec

In 2013, Mikolov and others put forward the Word2vec model, which achieved good results in many natural language processing tasks and attracted wide attention[4]. Due to the limitation of

calculation level and model structure, traditional text representation methods can only obtain the previous word of the current word as the relevant context of the current word, and predict the next word through the previous word. This method is easily disturbed, resulting in data deviation. Word2vec believes that many adjacent words in a sentence may have an impact on the current words. This method obtains the context information of the current words for training and constructs word vectors, which can eliminate the data instability caused by small samples and accurately represent the correlation between words. At the same time, word2vec method can reflect the similarity between word vectors in vector space, and can express rich semantic features with fewer dimensions, thus avoiding dimension explosion. Word2vec includes two models: CBOW(Continuous Bag-of-Word) and skip-gram. Skip-gram model predicts the surrounding words by the words in the current position; CBOW, on the contrary, is a word that predicts the current position by surrounding words.

CBOW first needs to determine the context window of the current word, that is, the first and last words of the current word.

$$P(W_t \mid W_{t-c} : W_{t+c}) \tag{1}$$

For a sequence of length, the goal is to maximize the following log-likelihood function:T

$$L = \frac{1}{T} \sum_{t=1}^{T} \log P(W_t \mid W_{t-c} : W_{t+c}) \tag{2}$$

Different from CBOW's training idea, skip-gram is to use a given word to predict the words that may be generated around it. The logarithmic likelihood function of this method is shown in the following formula:

$$L = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \le j \le c, j \ne 0} \log P(W_{t+j} \mid W_t) \tag{3}$$

C represents a window, w_{t+j} indicating the values before and after the window. Find the word sequence that maximizes the above formula.

2) GloVe

GloVe, called Global Vectors for Word Representation, is a word representation tool based on global word frequency. GloVe needs to construct Co-ocurrence Matrix, construct loss function, train GloVe model, and use the relationship between word vector and co-occurrence matrix to get word vector[5].

3. Language pre-training model

Distributed text representation method has the advantages of moderate dimension, ability to express the semantics of text, and combination of context, etc., and has achieved good results in the field of natural language processing. However, it is still a static text representation method. After the text vector is generated, it will remain fixed, and the vector will have a one-to-one relationship with words. Polysemous words will also be mapped into the same vector when expressing different semantics.

In view of this situation, the researchers put forward a pre-training model. Pre-training model is a network that has been trained and saved on a large data set. The model includes network structure and trained parameters. Subsequent use only needs to take the trained parameters as initial parameters, and fine-tune them on specific data sets. Even if the subsequent data set is not large in scale, a good

model effect can be obtained. Using this method can provide better prior information, avoid random initialization, provide faster training speed, and reduce the requirement for data volume[6].

3.1. BERT pre-training model

In 2018, Devlin and others proposed the bidirectional encoder representations from transformer (Bert) model. Subsequently, this model surpassed other models in a number of NLP tasks, and achieved state-of-art level results. This model uses multi-layer Transformer structure and attention mechanism, and after massive data training, it has strong text feature processing ability. To analyze the reason why BERT has strong ability to process text features, it is necessary to explore BERT's structure, Transformer encoder, self-attention mechanism, multi-head mechanism, BERT's training task and fine-tune thought[7].

BERT's structure: The idea of BERT algorithm is to do unsupervised training on a lot of unlabeled data at first, so as to obtain the internal logic rules of language. BERT is composed of three parts: Embedding layer, Transformer structure and loss optimization. The structure is shown in figure2. The input information includes token embeddings, semantic embeddings and position embeddings. BERT Embedding is the sum of three kinds of Embedding. A vector generated by embedding characters, with the [CLS] sign at the beginning of the sentence, indicating the beginning of the text. In BERT's design, in order to give consideration to NLP models such as question answering system, the input training data are two sentences in pairs, separated by [SEP] separator. Segement embeddings are used to distinguish between two input sentences. The words of the former sentence are expressed, and the words of the latter sentence are expressed. Position embeddings indicates the position information of words in the sequence, so that the model can make use of word order information. The length of all three embedding is 768[8].

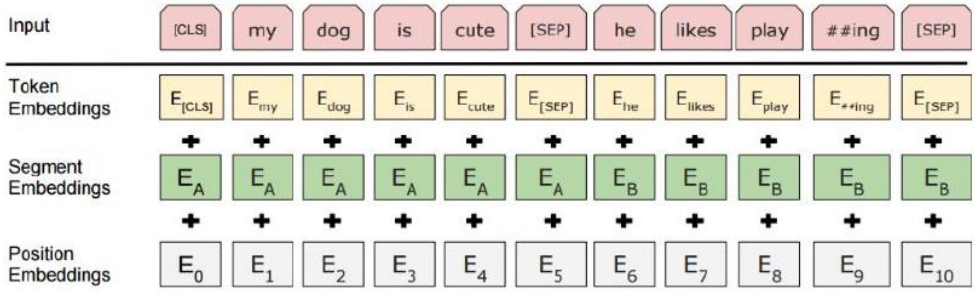

Figure 2: BERT Input Schematic Diagram

3.2. ALBERT pre-training model

Although BERT has excellent performance, its shortcomings are obvious. First of all, because BERT's parameters are huge, in use, it may exceed the memory size of the graphics card, resulting in time-consuming and inconvenient training. Moreover, BERT simply stacks many neural layers, but it can't increase the performance, resulting in the bottleneck of performance. Many researchers began to study how to compress BERT's parameters while keeping BERT's performance, so as to improve training and use efficiency. ALBERT proposed by Lanand others is a relatively perfect improved version, which can improve the efficiency and performance by reducing the parameters and increasing the depth of the model[9] . The improvement is mainly reflected in the following aspects:

Firstly, embedding mapping matrix, ALBERT decomposes the mapping matrix, thus reducing the scale of parameters to describe the mapping matrix. Secondly, ALBERT realizes cross-layer parameter sharing. ALBERT shared a parameter from the attention layer to each attention head, thus achieving the purpose of compressing the parameter quantity. And will not lose too much information.

These improvements make ALBERT's training speed double that of BERT under the same configuration.

In addition to the above improvements, ALBERT also optimized BERT's training task, upgrading the next sentence prediction task to sentence order prediction task. This method improves the model's cognition of the continuity between sentences, and has better processing ability for multi-sentence tasks.

4. Network structure of text vectorization representation method based on language pre-training model

The traditional word embedding model has limited ability to understand and express text because of its static, vector and word-to-word characteristics. This section introduces the language pre-training model to replace the traditional word embedding model and realize the dynamic word embedding function. RNN can extract the features of the context, and has a good understanding and processing ability of natural language. LSTM is based on RNN and improved. By introducing "forgetting gate" to solve the problem of gradient explosion, the model's ability to process text is strengthened. This section combines the advantages of language pre-training model and LSTM network to design the model.

4.1 Model structure

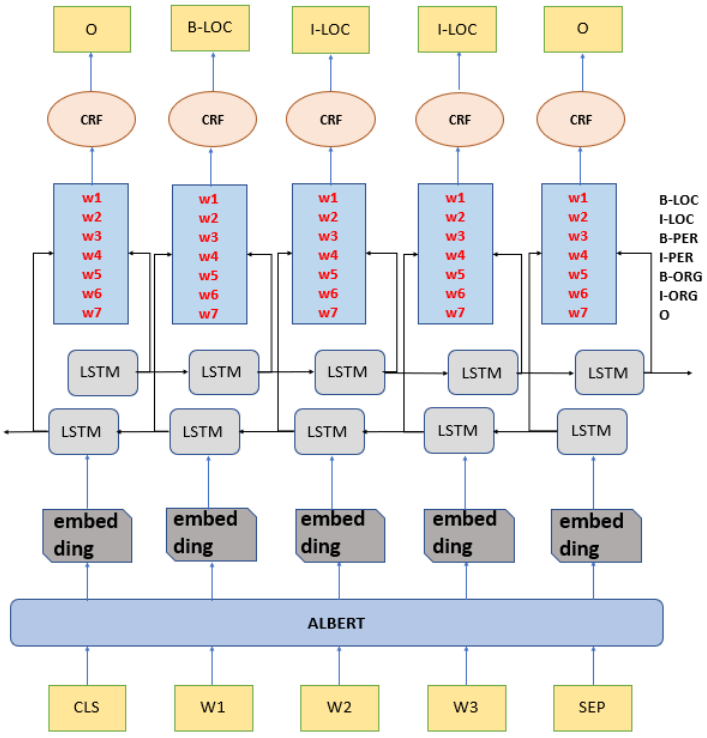

Figure 3: Model structure diagram

The model structure is shown in Figure 3. Firstly, ALBERT is used to vectorize the text, and word vectors are generated to express semantics and word features. Then, the word vector is input into the BiLSTM network, and the context information is fully extracted and learned by using the processing ability of the BiLSTM network. Then, the probability that the input characters belong to a certain category is judged one by one, and the probability matrix is generated. Then, the probability matrix

is input into CRF, and CRF selects the link with the highest probability from the probability matrix and outputs it. Thus, the sequence of category labels is obtained.

4.2. Functional modules

This section analyzes the functional structure of ALBERT and BiLSTM, the main modules of the model.

1) ALBERT model

The data set contains a large number of unknown words and abbreviations. Aiming at the defects that the traditional word embedding method can't solve the expression of polysemy and can't fully express the semantic features of the text, this paper introduces the language pre-training model into the field of named entity recognition for the first time. Using ALBERT model to represent text vectorization not only solves the problem of polysemy, but also makes the generated vector fully express the semantic features of words, thus enhancing the recognition ability of unknown words and abbreviations. ALBERT introduced several versions, including: the information of each version is shown in Table 1. Considering the factors such as training time, hardware conditions and model effect, this model uses the model.

Table 1: ALBERT Version and Parameter Table

model	Para_size	level	Hidden_level	Word_emb
$ALBERT_{BASE}$	12M	12	768	128
$ALBERT_{LARGE}$	18M	24	1024	128
$ALBERT_{XLARGE}$	59M	24	2048	128
$ALBERT_{XXLARGE}$	233M	12	4096	128

2) BiLSTM model

Recurrent neural networks are usually used to process texts with temporal characteristics. In the process of data transmission, the gradient of parameters is multiplied continuously, which is easy to cause the problem of gradient disappearance or gradient explosion, which affects the training, so researchers put forward the long-term and short-term memory neural network based on RNN. Long Short-term Memory Networks (LSTM) is a special RNN, which can solve the problems of gradient disappearance and explosion in long sequence training engineering. In order to make the model fully learn and utilize the context information of the text, especially the long-distance information, LSTM structure is added to the model. LSTM can capture long-distance sequence information, and it is powerful in modeling sequence data. This ability is inseparable from the cell structure of LSTM. At time T, the input state of LSTM unit includes memory cell, the previous hidden layer and input layer, and its main components include input gate, forgetting gate, output gate and memory cell. The corresponding formula is:

$$i_t = \sigma(w_{x_i} x_t + w_{h_i} h_{t-1} + w_{c_i} c_{t-1} + b_i) \tag{4}$$

$$f_t = \sigma(w_{x_t} x_t + w_{h_f} h_{t-1} + w_{c_f} c_{t-1} + b_f) \tag{5}$$

$$c_t = f_t c_{t-1} + i_t \tanh(w_{x_i} x_t + w_{h_c} h_{t-1} + b_c) \tag{6}$$

$$o_t = \sigma(w_{x_o} x_t + w_{h_o} h_{t-1} + w_{c_o} c_{t-1} + b_o) \tag{7}$$

The forward LSTM and the backward LSTM are combined into BiLSTM, so that the sentence information can be obtained from front to back and from back to front in both directions, the purpose of obtaining context information is achieved, and a better semantic expression effect is achieved. The structure of BiLSTM is shown in Figure 4:

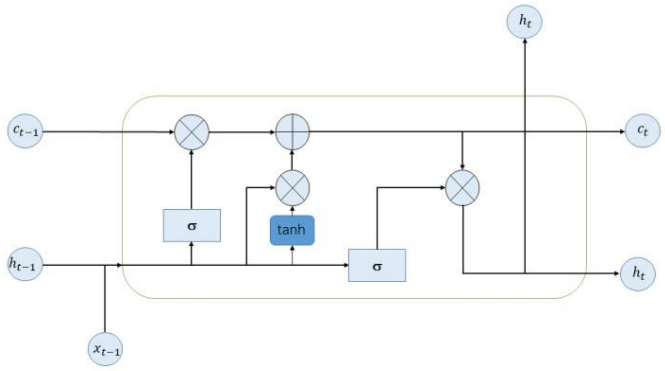

Figure 4: cell structure diagram of bilstm

5. Experiment and analysis

5.1. Experimental environment and parameters

Table 2: Computer Environment Settings Table

project	parameter
CPU	Core i7
GPU	Nvidia Geforce GTX 3200
system	Ubuntu 18.04
Python version	3.6
Tensorflow version	1.15

Table 3: Table of Superparameter Values

Superparameter	parameter
max_seq_length	128
dropout	0.1
Learning rate	2e-5
BiLSTM hidden layer	128
CRF_cost	0.2
CRF_eta	0.0005
optimizer	SGD
Epoch	100
Batchsize	64
ALBERT model	$ALBERT_{BASE}$

The training of deep neural network model has certain requirements for computer configuration. Using graphics card (GPU) can get faster calculation effect than CPU and shorten training time. In

terms of operating system, Ubuntu system is superior to Windows system in supporting model training. TensorFlow framework is used for deep learning, and Python3.6 is used for programming language. ALBERT version uses version. The configuration of the computer is shown in Table 2, and the settings of super parameters are shown in Table 3.

5.2. Experimental scheme

In the experimental group, ALBERT was used to complete the vectorization of the text, and then the vector was input into the BiLSTM model. After the BiLSTM model is fully learned and extracted, the label matrix is generated, which is input into CRF model and the prediction result is output. The control group uses the distributed word embedding method to complete the vectorization process of the text. Word embedding methods used include CBOW, Skip-gram and GloVe. Then, the word embedding results are input into the BiLSTM-CRF model, and the classification results are output.

5.3. Experimental results and analysis

The results on the People's Daily data set are shown in Table 4 and Figure 5. It can be seen that, compared with the traditional word embedding method, the model using language pre-training method has a 5.52%-5.88% improvement in accuracy, a 5.57%-6.41% improvement in recall and a 5.55%-6.15% improvement in F1 value. The results show that, compared with the traditional word embedding model, the model using language pre-training method has obvious positive effects on evaluation indexes such as P, R and F1. This shows that the method of text vectorization based on language pre-training model has a positive effect on the task of named entity recognition.

Table 4: Experimental Results of People's Daily Data Set

Model	P	R	F1
CBOW-BiLSTM-CRF	90.66	89.41	90.03
Skip-gram-BiLSTM-CRF	91.02	90.25	90.63
GloVe -BiLSTM-CRF	90.73	89.87	90.30
ALBERT-BiLSTM-CRF	96.54	95.82	96.18

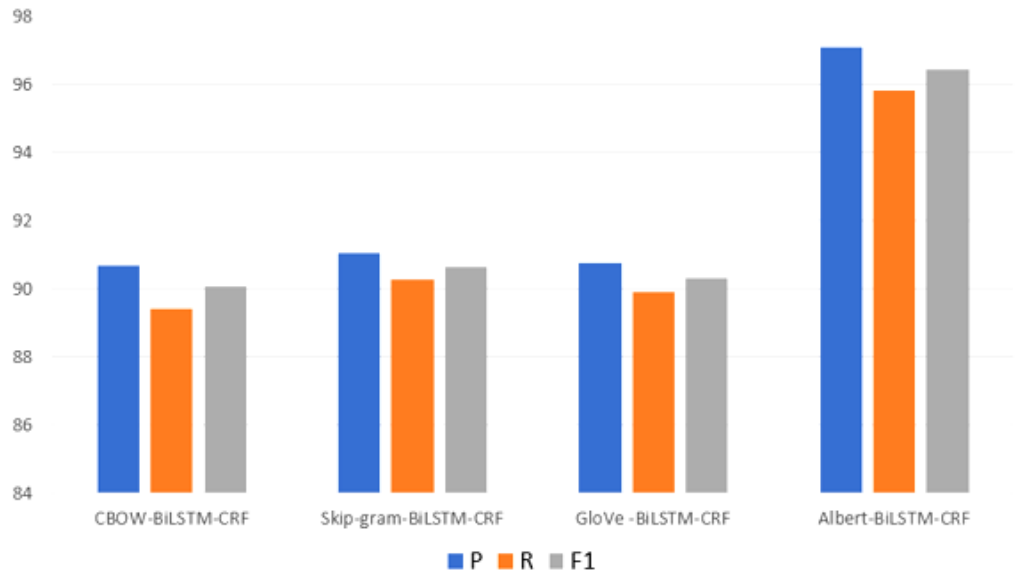

Figure 5: Experimental Results of People's Daily Data Set

6. Conclusions

Aiming at the problem that the existing models have insufficient ability to recognize unknown words and abbreviations commonly found in named entity recognition data, this paper proposes a text vectorization representation method based on language pre-training model. Firstly, this chapter introduces the vector representation of text, including discrete representation and distributed representation. Then, it analyzes the shortcomings of the traditional word embedding methods (word2vec, GloVe). Then, the language models based on pre-training, BERT and ALBERT, are analyzed. The analysis includes the structure of BERT model, the method of Transformer, training tasks, fine-tuning operation and ALBERT's improvement of BERT. Then, a text vectorization method based on language pre-training model is proposed to replace the traditional word embedding method. Finally, a comparative experiment is designed. On the data set of People's Daily, the text vectorization method based on language pre-training model is compared with word embedding methods such as word2vec and GloVe. By analyzing the results, the effectiveness of the proposed method is verified.

References

[1] Goelz S E, Hamilton S R, Vogelstein B. Purification of DNA from formaldehyde fixed and paraffin embedded human tissue [J]. Biochem Biophys Res Commun, 1985, 130(1):118-126.

[2] Kendall A, Cipolla R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation [J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2017:1-1.

[3] Bengio Y, Schwenk H, Jean-Sébastien Senécal, et al. Neural Probabilistic Language Models[J]. The Journal of Machine Learning Research, 2003, 3(6):1137-1155.

[4] Mateusz Szczepański, Pawlicki M, Kozik R, et al. The Application of Deep Learning Imputation and Other Advanced Methods for Handling Missing Values in Network Intrusion Detection [J]. Vietnam Journal of Computer Science, 2023, 10(01):1-23.

[5] Buczak A, Guven E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection [J]. IEEE Communications Surveys & Tutorials, 2017, 18(2):1153-1176.

[6] Pi X, Iijima B A, Lu W. Effects of Ionospheric Scintillation on GNSS-Based Positioning [J]. Navigation, 2017, 64(1):3-22.

[7] Chandrashekar S, Bashel B, Balasubramanya H, et al. UALCAN: A Portal for Facilitating Tumor Subgroup Gene Expression and Survival Analyses [J]. Neoplasia (New York, N.Y.), 2017, 19(8):649-658.

[8] Du J, Cheng K, Yu Y, et al. Panchromatic Image Super-Resolution Via Self Attention-Augmented Wasserstein Generative Adversarial Network [J]. Sensors, 2021, 21(6):2158.

[9] Markley J L, R Brüschweiler, Edison A S, et al. The future of NMR-based metabolomics [J]. Current Opinion in Biotechnology, 2017, 43:34-40.