Academic Journal of Computing & Information Science
ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

Entity recognition of Chinese bidding announcement title based on deep learning

Kong Zhang

College of Information Engineering,Nanjing University of Finance and Economics, 210023, China
418921061@qq.com

ABSTRACT. *Bidding announcement widely exists in Chinese government procurement websites at all levels in the form of text. Its composition is complex and the number is numerous. Identifying and extracting more concise project names is helpful to improve the ability of website data query and analysis. To solve this problem, this paper proposes a Transformer-att-label model based on deep learning. The model uses Transformer-att for feature extraction. It uses the attention mechanism to replace the original multi-head combination of Transformer, which can improve the recognition effect. And combined with tag embedding, predict the tag semantics of words, and select the tag closest to its semantics for output. The proposed model was tested on the announcement title data set of the Chinese bidding website, and the recognition effect of other mainstream models was compared to verify the effectiveness of the method.*

KEYWORDS: *Chinese project name, named entity recognition, deep learning, Transformer, label*

1. Introduction

The Chinese bidding website gathers a large amount of information scattered on the websites of various units, and has become one of the main transaction channels for the government and various enterprises and institutions to conduct commodity transactions, project construction, and service provision. Among them, the announcement title data information is concentrated, the format is diverse, and a large number of lengthy and complex titles make it difficult for users to read. Therefore, it is urgent to identify (extract) short and clear project names, place names and time from these announcement titles, which is more convenient Provide users with in-depth query functions to achieve a friendly user experience.

Named entity recognition (NER) is also called entity extraction, which aims to extract named entities from free text and classify the extracted named entities into certain categories. The earliest entities specified by MUC-6[1] include three major

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

categories (person names, place names, and time names), but now the broad definition is to identify meaningful or referential nouns in a certain field, and combine these The classification of entity words is of great significance to the study of many complex tasks. Traditional NER technologies are rule-based, such as the DLCo Train method proposed by Collins et al. [2], and statistics-based, such as Hidden Markov Model (HMM) [3], Maximum Entropy Model (ME) [4-5], Support Vector Machine (SVM) [6-7] and Conditional Random Field (CRF) [8-9] and based on mixing multiple methods, such as mixing multiple SVMs, combining HMM and ME, combining Combination of statistics and rules. With the emergence of distributed representation methods such as word embedding, deep learning [10-11] has also developed rapidly in the field of natural language processing. Researchers usually use Convolutional Neural Network (CNN) [12] or Recurrent Neural Network (RNN) and its variants [13] as feature extractors for feature extraction, and use CRF as annotation constraints for entity recognition. In recent years, the Transformer model using the self-attention mechanism has been extensively studied. Its powerful parallelism speeds up the efficiency of engineering, and the deeper network enhances the expression ability of the model, which can reduce the extraction of artificial features and improve the generalization ability of the model. Realize the end-to-end task.

Announcement title data has various forms, relatively complex context, and entities may belong to different types in different contexts. The common model cannot handle it well, so this paper proposes the Transformer-att-label model. Use Transformer-att for feature extraction, which uses traditional attention mechanism to combine multiple attention heads, so that the model can pay more attention to important aspects. In order to solve the impact of the accuracy of Chinese word segmentation on the effect of entity recognition, the word vector processed by CNN is used to supplement the characteristics of the word vector, so that the input contains more semantic information. It also considers the connection between tags, and predicts the most likely tag of a word by combining semantic embedding.

2. Model framework

As shown in Figure 1, the whole model framework is divided into three parts: input processing, feature extraction and joint annotation. First, perform word segmentation, use word2vec to train word vectors with semantic features, and use CNN to extract the features of each word to form a word vector, stitch the word vector and the word vector as input, and then use the Transformer-att model for feature extraction. Finally, combined with tag embedding, the possible tag semantics of words are predicted for labeling.

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

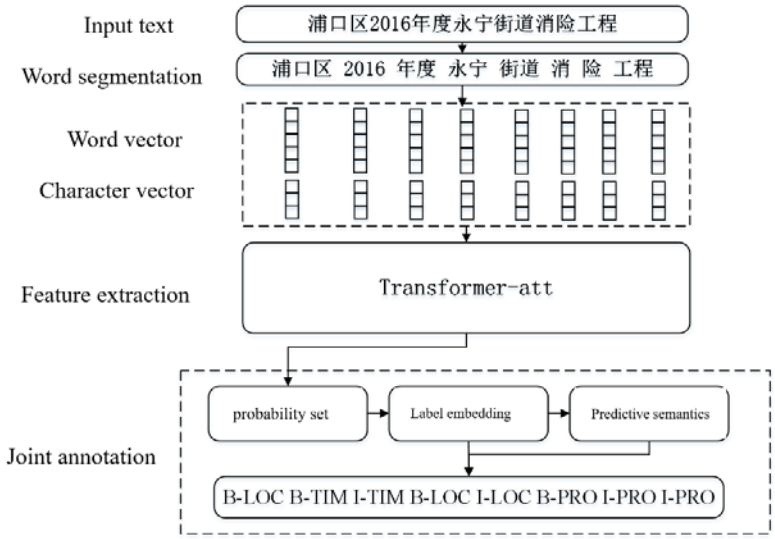

Figure. 1 Transformer-att-label model

2.1 Model input

Studies by Chiu and Nichols [15], Ma and Hovy et al. [16] have shown that some character-level morphological information can be obtained by processing word vectors with CNN. Therefore, in this paper, the word vector trained by word2vec and the word vector splicing processed by CNN are used as features for input. First, the character vector of each word is formed into a matrix as the input of CNN. For the problem of inconsistent word length, the longest length of a word can be set. For shorter words, padding is added to the left and right.

First, the character vector of each word is formed into a matrix as the input of CNN. For the problem of inconsistent word length, you can set the longest length of a word. For shorter words, add padding to the left and right, and then perform convolution. Extract the features between characters in a word, and finally perform a maximum pooling operation, select the most representative feature, and form a word vector for output. The splicing vector of the word vector and the word vector not only contains the feature of the word but also the feature between the characters, which contributes to the subsequent feature extraction.

2.2 Feature extraction

The Transformer model uses a self-attention mechanism to focus on the influence of other words before and after a word, and it combines the semantics of the context well. Let the input vector be $H = [h_1, \cdots, h_N] \in \mathbb{R}^{l \times d}$, $\mathrm{MultiHead}(H) \in$

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

$\mathbb{R}^{l\times d}$ is the final output of multi head combination, Where l is the length of the input sentence and d is the dimension of the input vector, the calculation formula is as follows:

$$Q^{(h)}, K^{(h)}, V^{(h)} = HW_q{}^{(h)}, HW_k{}^{(h)}, HW_v{}^{(h)} \quad (1)$$

$$A_{t,j}{}^{(h)} = Q_t{}^{(h)} K_j^{(h)\,T} \quad (2)$$

$$\mathrm{Attn}\left(Q^{(h)}, K^{(h)}, V^{(h)}\right) = \mathrm{SoftMax}\left(\frac{A^{(h)}}{\sqrt{d_k}}\right) V^{(h)} \quad (3)$$

$$Z^{(h)} = \mathrm{Attn}\left(Q^{(h)}, K^{(h)}, V^{(h)}\right) \quad (4)$$

$Q^{(h)}, K^{(h)}, V^{(h)}$are the query/key/value vector matrix of each attention head h, h is the index of multiple heads, the conversion matrix of each attention head is $W_q, W_k, W_v \in \mathbb{R}^{d\times d_k}$, d_k is a parameter, and its size is generally d/n, n is The number of longs. Q_t is the query vector of word t, K_j is the key vector of word j, and $A_{t,j}$ is the score of words t and j, that is, the influence of word j on word t. It uses the method of scaling dot product to solve the problem that the vector is too long, the scores are concentrated, and it is not easy to distinguish. $Z^{(h)}$ is the weighted calculation of the value vector of each word after normalization by SoftMax in each head.

$$\mathrm{MultiHead}(H) = \left[Z^{(1)}; \ldots; Z^{(n)}\right] W_O \quad (5)$$

The combination of multiple attention heads of the model is simple splicing and combination, as shown in formula (5), although the calculation results of multiple heads are combined, it cannot distinguish the importance of different heads. This paper proposes that the Transformer-att model adds a traditional attention mechanism when combining multiple heads, giving different weights to each head. It first maps the feature matrix of the multiple heads. The formula is shown in (6), and then the dot product is used. The model is scored:

$$Z'^{(h)} = Z^{(h)} W_z \quad (6)$$

$$a^{(h)} = \mathrm{SoftMax}\left(s\left(Z'^{(h)}, q\right)\right) = \frac{\exp\left(s\left(Z'^{(h)}, q\right)\right)}{\sum_{k=1}^{n} \exp\left(s\left(Z'^{(k)}, q\right)\right)} \quad (7)$$

Where $Z^{(h)}$ is the feature matrix of each attention head under the multi-head self-attention mechanism, and the dimension is $\mathbb{R}^{l\times d_k}$. W_z is the transformation matrix, the dimension is $\mathbb{R}^{d_k\times 1}$, and each head is mapped into a one-dimensional space. q is the query vector, and the dot product model is used to calculate the attention score of each head. The same goes through the SoftMax function to get the final score $a^{(h)}$ for each head.

$$\mathrm{MultiHead}(H) = \left[a^{(1)}Z^{(1)}; \ldots; a^{(h)}Z^{(n)}\right] W_O \quad (8)$$

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

Finally, the score of each attention head is used as the weight and each head is multiplied and then spliced as the output. This solves the problem of different impacts of different attention heads, and makes the model focus on the aspects of large impact in different situations, and optimizes the final Output.

2.3 Joint annotation

Due to the lack of supervised data and insufficient model feature extraction capabilities, there are not enough features in the data to represent certain tags. This paper proposes a Transformer-att-label method with tag embedding, which embeds tags in the semantic space to find the relationship between tags. First use Transformer to calculate the probability that the input vector x is labeled as label $y \in \mathcal{Y}$, expressed as $p(y|x)$, and $\sum_{y=1}^{n} p(y|x) = 1$. Let $\hat{y}(x, 1)$ denote the most likely label based on the output of the vector x input by the marker, which is expressed in form as:

$$\hat{y}(x, 1) \equiv \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, p(y|x) \tag{9}$$

The similar $\hat{y}(x, t)$ represents the t-th most likely label of the input vector x, that is to say $p(\hat{y}(x, t)|x)$ is the t-th largest value in $p(y|x); y \in \mathcal{Y}$;y\in\mathcal{Y}. Given the highest t predictions of the input x, the model deterministically embeds the predicted label semantics of the input x into the vector $f(x)$ as the corresponding probability weight of the label semantic embedding $s(\hat{y}(x, t))$. The formula is:

$$f(x) = \frac{1}{Z} \sum_{t=1}^{T} p(\hat{y}(x, t)|x) \cdot s(\hat{y}(x, t)) \tag{10}$$

Where Z is the normalization factor $Z = \sum_{t=1}^{T} p(\hat{y}(x, t)|x)$, where T Is the hyperparameter of the maximum number of tags to be considered. If the tagger predicts the label y of x very surely, that is, $p(y|x) \approx 1$, then $f(x) \approx s(y)$, but if The tagger is not sure which tag it is, so it will be closer to the tag with higher probability in the semantic space of the tag. Given the expected embedding of x in the semantic space, that is, $f(x)$, the label can be found by finding the embedding closest to $f(x)$ in the semantic space, and the embedding vector can be sorted using cosine similarity to calculate the label $\hat{y}(x, 1)$ with the highest similarity. The formula is:

$$\hat{y}(x, 1) \equiv \underset{y \in \mathcal{Y}}{\operatorname{argmax}}\, \cos(f(x), s(y)) \tag{11}$$

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

3. Experiment and analysis

3.1 Data preprocessing

The data of this experiment comes from the announcement title of the Internet China Bidding and Purchasing Network. A total of 280,950 pieces of data were obtained. The jieba word segmentation with better word segmentation effect was used, and the BIO method was used for marking. The marking method is shown in Table 1. After labeling, we finally selected 10,000 pieces of data as experimental data, including 19,382 ground nouns, 5823 time words, and 12,843 engineering words as experimental data. Among them, 70% of the data is used as the training set training model parameters, 10% is used as the validation set to verify whether there is a problem with the model learning effect, and 20% is used as the test set to test the recognition result. The rest of the data will be used as corpus for word2vec training.

Table 1 Experimental data of sensor measurement accuracy

B-LOC	Place word begins	I-LOC	Place word middle
B-TIM	Time word begins	I-TIM	Time word middle
B-PRO	Engineering word begins	I-PRO	Engineering word middle
O Not belonging to the word to be recognized			

3.2 Lab environment

The main parameters of the experimental environment used in this article are operating system: Windows10; processor: Intel(R) Xeon(R) W-2102 @ 2.9GHz; memory: 16G; programming: python 3.6; built using Google's deep open source framework TensorFlow 1.11.0 All neural network models are trained and tested

3.3 Result analysist

The evaluation indicators of this experiment are correct rate (P), recall rate (R), and F1 value, which are used to evaluate system performance. Calculated as follows:

$$P = \frac{\text{Correctly identify the number of entities}}{\text{Total number of identified entities}} \tag{12}$$

$$R = \frac{\text{Correctly identify the number of entities}}{\text{The number of entities in the document}} \tag{13}$$

$$F_1 = \frac{2 \times P \times R}{P+R} \tag{14}$$

In order to verify the effect of the model proposed in this paper, the experiment uses the Transformer-CRF model, the Transformer-att-CRF model, and the Transformer-att-label model to conduct experiments, and compares them with the mainstream model BiLstm-CRF. The experimental results are as follows:

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

Table 2 Transformer-CRF experiment results

Entity category	P	R	F1
Place word	0.8445	0.8426	0.8435
Time word	0.9324	0.9074	0.9197
Engineering word	0.7364	0.7935	0.7638
average value	0.8377	0.8478	0.8423

Table 3 Transformer-att-CRF experiment results

Entity category	P	R	F1
Place word	0.8555	0.8514	0.8534
Time word	0.9314	0.9125	0.9219
Engineering word	0.7882	0.7923	0.7902
average value	0.8584	0.8521	0.8552

From the experimental results in Table 2 and Table 3, it is more appropriate to use attention combined with multiple heads in the Transformer model. The F1 value of the original model is improved by 1.0% in the recognition of place names, and the time is improved. 0.2%, an increase of 1.6% in the project name. Among them, the accuracy of project names has increased by 3.1%, which shows that multi-head attention is very helpful in understanding the meaning of complex words, and the combination of good or bad combinations can reduce the probability of identifying wrong entity words.

Table 4 Transformer-att-label experiment results

Entity category	P	R	F1
Place word	0.8628	0.8545	0.8586
Time word	0.9424	0.9374	0.9398
Engineering word	0.7464	0.8124	0.7780
average value	0.8505	0.8681	0.8588

Table 5 Transformer-att-label experiment results

Entity category	P	R	F1
Place word	0.8628	0.8545	0.8586
Time word	0.9424	0.9374	0.9398
Engineering word	0.7464	0.8124	0.7780
average value	0.8505	0.8681	0.8588

From Table 1-4 and Table 1-5, after adding the tag embedding method, various recognition effects are improved, especially compared with the current mainstream general model BiLstm-CRF. The F1 value in place name recognition increased by 3.87%, the F1 value in the recognition of time words was 2.1% higher, and the F1 value in the recognition of engineering nouns was 3.1% higher. It shows that the Transformer-att-label model with tag embedding has greatly improved the

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

classification of entity words in the recognition process, and is more suitable for processing complex data, especially in determining the most difficult place names and project names.

4. Conclusion

This paper proposes the Transformer-att-label model for the centralization, diverse forms and complex entity types of announcement header information. This model improves the combination of multiple attention heads and uses the traditional attention mechanism for multi-head fusion, which makes the model pay more attention to important attention heads and improves the attention effect of multi-head attention. In addition, tag embedding is added for joint annotation, which makes the model's classification of entity words more accurate. Finally, the commonly used model and the Transformer model that has not been improved are used for comparative experiments. Experiments have proved the effectiveness of the model, which has improved F1 values of all categories.

References

[1] Chinchor N (1995). MUC-6 named entity task definition (version 2.1) [C]. Proceedings of the 6th Conference on Message Under-standing, Columbia, Maryland.

[2] Collins M, Singer Y (1999). Unsupervised models for named entity classification [C]. Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora: 100-110.

[3] Zhou G D, Su J (2002). Named entity recognition using an HMM-based chunk tagger[C]. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics: 473-480.

[4] Chieu H L, Ng H T (2002). Named entity recognition: a maximum entropy approach using global information [C]. In: Proceedings of the 19th international conference on Computational linguistics. Association for Computational Linguistics: 1-7.

[5] Borthwick A, Grishman R (1999). A maximum entropy approach to named entity recognition [D]. New York University, Graduate School of Arts and Science,:1-11.

[6] Isozaki H, Kazawa H (2002). Efficient support vector classifiers for named entity recognition [C]. Proceedings of the 19th Interna-tional Conference on Computational Linguistics. Stroudsburg: Association for Computational Linguistics, 1: 1-7.

[7] Ekbal A, BandyopadhyayS (2010). Named entity recognition using support vector machine: A language independent approach [J]. International Journal of Electrical, Computer, and Systems Engineering, 4 (2): 155-170.

[8] Lafferty J, McCallum A, Pereira F C N (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data [J].:1-15.

ISSN 2616-5775 Vol. 3, Issue 3: 83-91, DOI: 10.25236/AJCIS.030309

[9] McCallum A, Li W (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons [C]. In: Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003. Association for Computational Linguistics, 2003: 188-191.
[10] Le Cun Y, Bengio Y, Hinton G (2015). Deep learning [J]. Nature, 521 (7553): 436-444.
[11] Hirschberg J, Manning C D (2015). Advances in natural language processing [J]. Science, 349 (6245): 261-266.
[12] Pinheiro P H O, ColloBert R (2013). Recurrent Convolutional Neural Networks for Scene Parsing [J].
[13] Peng N, Dredze M (2016). Improving named entity recognition for Chinese social media with word segmentation representation learning [C]. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Stroudsburg: Association for Computational Linguistics 149-155.
[14] Zhou G D, Su J (2002). Named entity recognition using an HMM-based chunk tagger[C]. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics. Association for Computational Linguistics,: 473-480.
[15] Chieu H L, Ng H T (2002). Named entity recognition: a maximum entropy approach using global information [C]. In: Proceedings of the 19th international conference on Computational linguistics. Association for Computational Linguistics,: 1-7.