IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Traitement du Signal
Vol. 39, No. 5, October, 2022, pp. 1833-1839
Journal homepage: http://iieta.org/journals/ts

Recognizing Chinese Predicate Heads Based on Textual Bounding Boxes

Lijuan Liu[1,2*], Xiao Guo[3]

[1] College of Physical Education, Guizhou Education University, Guiyang 550018, China
[2] Li Qiong Studio, Guizhou Education University, Guiyang 550018, China
[3] College of Computer Science and Technology, Guizhou University, Guiyang 550025, China

Corresponding Author Email: liulijuan@gznc.edu.cn

https://doi.org/10.18280/ts.390543

Received: 10 June 2022
Accepted: 25 August 2022

Keywords:
predicate head, textual bounding box, multi-object leaning

ABSTRACT

Predicate head plays a central grammatical role to organize relevant syntactic elements of a sentence. Identifying the predicate is the key to understanding a sentence. However, due to the characteristics of Chinese, recognizing Chinese predicate heads is a challenging task. This paper proposes a textual bounding box-based deep architecture for Chinese predicate recognition. In this method, a sentence is first transformed into an abstract representation. Then, textual bounding boxes are generated from the abstract representation, where a bounding box represents an abstract representation of a possible predicate head. Finally, an end-to-end multi-object learning framework is designed to learn the classification confidence and the offset of a bounding box relative to a true predicate head, which simultaneously predicts the class probability and locates predicate in a sentence. Experiments show that our model achieves competitive performance in a public Chinese predicate evaluation corpus. The result shows our model has the ability to enhance model discriminability and enable more potent nonlinear function approximators for Chinese predicate heads recognition.

1. INTRODUCTION

A predicate head (or predicate in short) is a verbal expression that plays a role as the structural center of a sentence. It is the central grammatical role to organize relevant syntactic elements in a sentence, e.g., subject, object, etc. Therefore, identifying the predicate is the key to understanding a sentence. Unlike many synthetic languages, e.g., English, they have affluent inflections to express syntactic relationships between words, recognizing predict head in Chinese is difficult. There are six distinctive characteristics, which makes the task to identify Chinese predicates more challenging.

First, Chinese is an ancient hieroglyphic. Chinese characters (or Hanzi) are ideograms where every character can convey a specific concept. Chinese hosts a large number of characters. Every Chinese character can be used as a word or as a morpheme, which considerably increases the searching space for predicate recognition. Second, a Chinese sentence is a sequence of characters, where a Chinese word is consisted of one or more characters without delimitation between them. Identifying predicates suffers from serious segmentation ambiguities. Third, Chinese verbs are usually multi-categorical in terms of part-of-speech (POS), without morphology indicating their verbal usages. Fourth, the predicate head in a Chinese sentence depends on its syntactic features of words. However, because Chinese is a hieroglyphic language, it is unable to depend on form and tense of words. Fifth, Chinese verbal expressions are often dynamically generated by rules or patterns. It is impossible to hold a predefined verb dictionary. Sixth, a Chinese sentence often contains several verbs. For example, in the one month of People's Daily corpus [1], 30% sentences are manually annotated with successive verbs. Distinguishing predicate from verbs is ambiguous, because each of these verbs can be considered as a predicate or as an adverbial phrase. Therefore, recognizing predicates in a Chinese sentence is a challenging task.

Recognizing Chinese predicates can be modeled as a sequence labeling task, where every character has predicated a tag to indicate its role in a predicate. For example, in the B-"I-O" encoding, tags of a character "B", "I" and "O" indicates that it is a start, inside or outside of a predicate. Given a sentence as input, a sequence model (e.g., Hidden Markov Model (HMM) [2, 3], Conditional Random Field (CRF) [4], or Long Short-Term Memory (LSTM) [5]) outputs a maximized label sequence for identifying predicates. Because the predicate is defined as the structural center of a sentence, a sentence usually contains only a predicate. Therefore, recognizing predicates is heavily depending on the global information of a sentence. The HMM and CRF often assume a first-order Markov dependency. They recognize a label element mainly depends on local features, which cannot ensure the output of a predicate. The LSTM uses a cell to memory the dependency information. However, when the distance is longer, the dependency information has vanished seriously.

In this paper, we propose a textual bounding box approach to support Chinese predicate recognition. In this approach, a sentence is first mapped into an abstract representation known as feature maps. Then, textual bounding boxes are generated from feature maps. A textual bounding box is an abstract representation of a possible predicate candidate. Each box has three parameters to indicate its class tag and its positions in a sentence. Because feature maps are generated by techniques such as Bi-LSTM [6] or Transformer [7], they encode semantic dependency information about a whole sentence. In the training process, for every box, in addition to predict the classification confidence, the position offset between the box and a true predicate are learned simultaneously. Compared with related works, our model has the advantage to make full use of annotation information and share parameters in the bottom of a deep neural network, which enhance model discriminability and enables more potent nonlinear function approximators for Chinese predicate heads recognition.

2. RELATED WORK

The predicate plays a central role in organizing syntactic or semantic information in a sentence, such as subject, object, time, etc. Identifying predicates is the key to understand a sentence. Previous works mainly adopt rules-based methods or statistical learning methods. In recurrent years, neural networks were widely used to recognizing predicates.

In rule-based methods, Luo and Zheng [8] analysed grammatical characteristics of predicates. A rule based method was proposed for predicate recognition, where rules are designed to identify predicates and boundaries. Li and Meng [9] proposed a method, which uses the syntactic relationship between subjects and predicates to support predicate recognition. In this model, comparing with previous works, syntactic analysis is more emphasized. In addition to static and dynamic grammatical features of a possible predicate, the interaction between rules is also taking into consideration.

In statistical learning methods, Chen and Shi [10] analysed the distribution of predicates in a corpus, which contains about 500,000 words. Then, a statistical based method was designed to identify predicates. Wang and Zhou [11] presented a maximum entropy (ME) method, where combined features about predicate are employed to make a prediction. Chen [12] proposed a probabilistic model for predicates recognition in which the context of predicates is analysed. Then, contextual characteristics that influence the occurrence of the predicates are selected for locating predicates in a sentence. Han et al. [13] proposed a method that combines lexical and syntactic features. Then, a C4.5 algorithm is adopted for predicate identification, which shows clear improvement.

There are also methods combine rules with statistical learning. For example, Gong et al. [14] divided the process to recognize predicate into three stages: phrase binding, predicate coarse screening and predicate fine screening. The main problem for this method is that sentences with complex structure cannot be correctly processed. As discussed by Ren et al. [15], the process to recognize predicates also divided into three stages. In the first preprocessing stage, maximal noun phrases in a sentence are replaced by “NP” labels to simplify the sentence structure. Then, predicates are recognized by a CRF model. Finally, in the post-processing stage, rules are designed to correct the preliminary recognition results.

In traditional methods, manually designed features are extracted from a sentence. Then, shallow algorithms (e.g., ME [11], CRF [15]) are employed to make a prediction based on categorical features. In recent works, neural networks were introduced to support predicate recognition. It has the advantage to automatically learn abstract features from raw inputs and encode semantic information by unsupervised methods. In this aspect, Li et al. [16] used large-scale Tibetan language corpus to train Tibetan word vectors, which significantly improved the Tibetan predicate recognition. Li et al. [17] presented a deep architecture, where an attention-BiLSTM-CRF model is used to extract possible predicates from a sentence. Then, to satisfy the uniqueness of predicates, a convolutional neural network is adopted to select the most likely predicate. Huang et al. [18] stacked a multi-layer BiLSTM network and a Highway network for sequence labelling. The output path was optimized by a constraint layer which was designed to reduce the influence of semantic vanishing problem.

Because predicate recognition uses the same techniques as named entity recognition, we also introduce related works about named entities recognition. Named entity recognition usually adopts sequence annotation model for recognition, such as HMM [2, 3], CRF [4], and LSTM [5]. In recent years, deep learning models have been widely studied. For example, Chen et al. [19] proposed a shallow recognition model based on boundary combination. It first recognizes the beginning and ending boundaries of entities, combines them into candidate entities, and then uses a classifier to make a prediction. Wang et al. [20] proposed a model, which maps a sentence to a pyramid like sequence layer, in which the recognition result of nested entities is taken as the output of the pyramid model. Yang and Tu [21] defined named entity recognition as an entity span sequence generation problem, and used the BART model with pointer mechanism to process the named entity recognition task. Shen et al. [22] proposed a joint task of boundary regression and span classification by locating entities first and then marking two-phase identifiers. Chen et al. [23] also proposed a boundary regression model, which also has the ability to predict entity types and locate named entities in a sentence. Compared with named entity recognition, predicate recognition emphasizes the grammatical function of predicates as the center of the sentence. The recognition is more dependent on the overall structure and semantic characteristics of the sentence.

3. MODEL

Before given the architecture of our model, definitions about textural bounding boxes are discussed as follows.

3.1 Textural bounding boxes

Let $X_t = [x_1, x_2, \dots x_N]$ be a sentence, where $x_i (1 \leq i \leq N)$ is a Chinese character. A neural network is adopted to transform X_t into a hidden representation denoted as $H_t = [h_i, h_2, \dots, h_N]$. The neural network to transform X_t into H_t is called a **basic network**, which can be truncated from a standard sentence labeling architecture for named entity recognition or POS tagging (e.g., a traditional BiLSTM architecture). In the basic network, many techniques (e.g., LSTM, Attention) can be used to capture global or dependency information of a sentence. In this paper, the hidden representation H_t is referred to as **feature maps**. Therefore, a feature map h_i is a high-order abstract representation of x_i. The representation can encode semantic dependency and global information by a deep neural network.

A textual bounding box d_i is defined as a subsequence of H_t denoted as $d_i = [h_s, \dots, h_{s+l}] (1 \leq s \leq s + l \leq N)$. A textual bounding box d_i can be represented as a triple $\langle s_j, l_j, c_j \rangle$. It has three parameters representing its class type, start position and length of the bounding box. Given a bounding box d_i, if the corresponding input characters $[x_s, \ldots, x_{s+l}]$ is a true predicate, the box is named as a **grounding truth bounding box**, referred as $g_j = \langle s_j, l_j, c_j \rangle$.

Given a sentence X_t and its feature maps H_t, let D_t represent all bounding boxes generated from H_t. It is represented as:

$$\mathrm{D}_t = \{d_i \mid d_i = \langle s_i, l_i, c_i \rangle, 1 \le s_i \le s_i + l_i \le n\} \tag{1}$$

Because every bounding box has parameters indicating its position in a sentence. Therefore, the ratio of intersection over union (IoU) between bounding boxes and a grounding truth box can be adopted to generate positive bounding boxes and negative bounding boxes. Let A and B represent the location ranges of two bounding boxes in a sentence. The IoU between A and B is computed as:

$$IoU(A,B) = \frac{A \cap B}{A \cup B} \tag{2}$$

All positive bounding boxes have a high overlapping ratio relative to some grounding truth bounding boxes. Therefore, positive bounding boxes contains effective semantic information for position regression. Negative bounding boxes are often far away from any truth box. They are mainly used to train a classifier for distinguishing true and false predicate instances.

Let G_t represents a truth bounding box set of X_t. The bounding box set D_t can be.

$$\begin{aligned} \mathrm{D}_t\mathrm{P} &= \{d_t \mid d_i \in \mathrm{D}_t, \exists g_j \in G_t(IoU(d_i, g_j) \ge \gamma)\} \\ \mathrm{D}_t\mathrm{N} &= \{d_i \mid d_i \in \mathrm{D}_t, \exists g_j \in G_t(IoU(d_i, g_j) < \gamma)\} \end{aligned} \tag{3}$$

The D_tP and D_tN are referred to as the positive bounding box set and the negative bounding box set.

In most of time, D_tP contains a few bounding boxes, but there is a large number of negative bounding boxes in D_tN. This leads to a serious data imbalance problem, which influences the performance and increase the computation complexity. Therefore, the ratio of intersection over union (IoU) between bounding boxes and a grounding truth box can be adopted to remove bounding boxes that are unlikely. This strategy is helpful for avoiding the data unbalance problem. In the training data, negative bounding boxes with lower IoU values are removed.

3.2 Training objective

Given a bounding box $d_i = \langle s_i, l_i, c_i \rangle$, the true class type c_i is a one-hot representation, where $c_i = [0, 1]$ and $c_i = [1, 0]$ means that it is a false predicate instance and a true predicate respectively. The true position offset between d_i relative to a grounding truth bounding box g_j is normalized as following:

$$\hat{s}_{ij} = (s_j - s_i)/s_j \qquad \hat{l}_{ij} = \log(l_j / l_i) \tag{4}$$

In this paper, a softmax layer and a linear layer are adopted to predict its classification confidence and to learn the position offsets relative to an adjacent grounding truth bounding box. The output of the softmax layer is denoted as $\tilde{c} = (\tilde{c}_0, \tilde{c}_1)$ in with $\tilde{c}_0$ and $\tilde{c}_1$ are the probabilities to be a negative predicate and a positive predicate. The output of position offset between d_i relative to a grounding truth bounding box g_j is represented as $(\Delta s_{ij}, \Delta l_{ij})$. Therefore, the total loss includes the location loss $(L_{loc}(x, s, l)$ and confidence loss $(L_{con}(x, c))$. It is represented as following:

$$L(x,s,l,c) = L_{loc}(x,s,l) + \alpha L_{con}(x,c) \tag{5}$$

The parameter α is used to balance the weight between L_{loc} and L_{loc}.

Given a bounding box $d_i \in D_tP$ and a truth box $g_j \in G_t$, let $E_{ij} = \{0,1\}$ be a characteristic function defined as follows:

$$E_{ij} = \begin{cases} 1, & \gamma \le \operatorname{argmax}_{g_i \in G_t} IoU(d_i, g_i) \\ 0, & others \end{cases} \tag{6}$$

Given a sentence X_t with its corresponding truth box set G_t and positive bounding box set D_tP, the location loss is computed as follows:

$$L_{loc}(x,s,l) = \sum_{g_j \in G_t} \frac{1}{N} \left(\sum_{d_i \in \mathrm{D}_t\mathrm{P}} E_{ij} (|\Delta s_{ij} - \hat{s}_{ij}| + |\Delta l_{ij} - \hat{l}_{ij}|) \right) \tag{7}$$

In Eq. (7), N denotes to the number positive bounding boxes that match to $g_j \in G_t$. The coefficient 1/N is used normalize the weight between different truth boxes. In the learning process, minimizing the $L_{loc}(x, c)$ is used normalize the weight between different truth boxes. In the learning process, minimizing the $L_{loc}(x, s, l)$ can result in every d_i approaching to the nearest truth box g_j.

The confidence loss $Lcon(x, c)$ is a cross-entropy loss that is formalized as following:

$$Lcon(x,c) = -\sum_{d_i \in \mathrm{D}_t\mathrm{P}} E_{ij} \log(\tilde{c}_1) - \sum_{d_i \in \mathrm{D}_t\mathrm{N}} \log(\tilde{c}_0) \tag{8}$$

The *training objective* is to reduce the total loss of the location offset and class prediction. In the training process, we optimize their locations to improve their matching degree and maximize their confidences.

3.3 Architecture of model

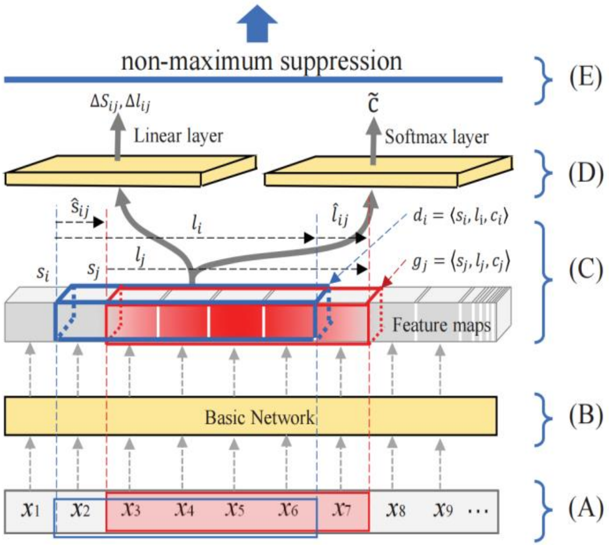

non-maximum suppression
$\Delta S_{ij}, \Delta l_{ij}$
$\tilde{c}$
Linear layer
Softmax layer
$\hat{s}_{ij}$
l_i
$\hat{l}_{ij}$
$d_i = \langle s_i, l_i, c_i \rangle$
l_j
$g_j = \langle s_j, l_j, c_j \rangle$
s_i
s_j
Feature maps
Basic Network
x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 ···
(E)
(D)
(C)
(B)
(A)

Figure 1. Architecture of model: (A) *input layer,* (B) *basic network,* (C) *bounding box generation,* (D)*output layer,* (E) *non-maximum suppression*

In this paper, we design an end-to-end multi-objective learning framework to simultaneously locate predict and the class probability. The architecture of this model is shown in Figure 1.

The architecture of the neural network is divided into five parts. Each part is discussed as follows.

(A) The input of this model is a sentence. In order to guarantee the same length for every sentence in the feature map layer, the length of sentence is set as 100. Longer sentences and shorter sentences are trimmed or padded respectively.

(B) A basic network is adopted to transform the input into feature maps. The basic network can be truncated from a standard sentence labelling architecture. In this paper, it is composed of an embedding layer and a Bi-LSTM layer. In the embedding layer, every input x_i is embedded into a distributed representation. The embedding can encode dense semantic information of words by pretrained lookup table. In this model, the BERT [24] approach is adopted to support the word embedding, which transforms every word into a 768 dimensional vector. The BiLSTM is a recurrent network that consists of two LSTMs [6]. It takes the input in a forward direction and a backward direction. In each direction, a cell is used to memory the previous hidden states. Therefore, the BiLSTM is effective to learn the semantic dependency in a sentence.

(C) Outputs of the basic network are named as *feature maps*. They are abstract representations of a sentence. Because the recurrent work is adopted to implement feature mapping, a feature map encodes semantic information of a word and its context. Let $[h_1, h_2, \dots h_N]$ denotes to the feature maps. According to Eq. (1), every feature map h_i is assembled with right feature maps for generating a default bounding box set $\{[h_i, h_{i+1}, \dots, h_{i+k}] | 1 \leq k \leq K\}$. In our experiment, 6 is set as the default value of K. Except the rightmost feature maps, it generates 6 default bounding boxes for each feature map. For each input sentence, there are 285 default bounding boxes in total. It generates a large number of negative bounding boxes, which leads to a serious imbalance problem. Eq. (3) can be used to filter bounding boxes. To reduce the computation complexity, the number of elements in D_tP and D_tN is limited to 1:3.

(D) A bounding box is an abstract representation of a possible predicate. Because Bi-LSTM is adopted to generate feature maps, every bounding box also contains the context information. Based on bounding boxes, a linear layer and a softmax layer are set to learn the classification confidence and the offset of a bounding box relative to a true predicate. It is a multi-object leaning framework, which reduces the total loss of the location offset and class prediction. In order to support the regression operation, values about bounding positions are normalized into interval [1, 0] to smooth the learning process.

(E) In the testing process, bounding boxes will approach to the truth bounding boxes, which leads to overlapped boxes near a truth bounding box. Non-maximum suppression searches local maximized boxes from overlapping neighborhoods. It suppresses elements that are not maximum values. In this paper, an 1D-NMS algorithm is used to select bounding boxes that are most likely true. The algorithm has three steps. First, a boundary box with the highest classification confidence is collected from the whole output. Second, the collected box is used to filter boundary boxes with the same type and overlapping ratio bigger than a predefined value ($\lambda \geq 0.65$). Third, collect another boundary with the highest classification confidence, and repeat the second process until all boundary boxes are processed.

4. EXPERIMENTS

Our method is evaluated based on a Chinese predicate corpus. It contains 762 documents collected from an official website on legal documents - China Judgments Online. The corpus is following the annotation guideline proposed by Chen et al. [25]. In the following, before evaluating the bounding box method, we first introduce the data set and experimental settings. Experiments show that bounding boxes can approach the true predicates in the training process.

4.1 Data set

According to characteristics of Chinese language, the key issue about annotating a predicate is that it acts as the center of a sentence, which organizes relevant syntactic elements. In the annotation guideline, several linguistic rules have been proposed to guarantee its structural central role in a sentence. First, instead of annotating a sentence as a tree structure, a flattened structure is adopted to represents the semantic role of a predicate in a sentence. Second, because commas (",") is ambiguous for separating Chinese sentences, the sentence boundary is manually annotated in the corpus. Third, the semantic roles of conjunctions in a sentence are distinguished to avoid ambiguity. Fourth, multi verb-noun phrases are proposed to guarantee the center role of predicates.

Chinese is a hieroglyphic writing system based characters (or Hanzi), where Chinese verbal expressions are often dynamically generated by rules or patterns. A verbal expression usually consists of several verbs. Because they act as a independent syntactic role in a sentence, the guideline handles them as a predicate. In the corpus, predicates are divided into five structures. (1) Singleton structure: a predicate consists of a transitive verb or intransitive verb. (2) Reduplicated structure: compound words generated by reduplicated methods, e.g., AA, AAB, ABB, AABB. (3) Coordinated structure: a predicate consists of coordinated verbs, which express relevant semantic meaning. (4) Modified structure: Verbs with modifiers, aspect markers and complements. (5) Include verbal expressions, e.g., proverbs, idioms, argots, allusions, etc.

In the corpus, there are 7022 sentences in total, where 7022 predicates have been annotated in the corpus. Each sentence has been annotated with the predicate and relevant syntactic elements (e.g., subject, object). The corpus is available at: https://github.com/YPench/Predicate-Head-Corpus.

4.2 Experimental settings

The evaluation documents are divided into 6:2:2 for training, validation, and testing. The precision/recall/F1-score (P/R/F) measurement is adopted to evaluate the performance. In the experiment, parameters of the neural network are shown in Table 1.

The embedding dimension is the output of BERT$_{BASE}$ [24], which maps every character into a 768 dimension vector. The LSTM dimension is the output of Bi-LSTM. Each LSTM outputs a 100 dimension vector at each time step. IoU threshold is value of γ in Eq. (3). Sample ratio is the ration between positive and negative bounding boxes. The NMS coefficient (λ) is used to filter bounding boxes from overlapping neighborhoods.

Table 1. Parameter settings

Parameter	Value	Parameter	Value
Batch size	30	Learning rate	0.001
Iterations	100	Optimizer	Adam
Embedding dimension	768	LSTM dimension	100
IoU threshold	0.5	Sample ratio	1:3
Sentence length	100	NMS coefficient	0.55

4.3 Evaluation

To evaluate the performance of bounding box based predicate recognition, we compare our model with five related works: CRF [16], LGN [26], Bi-LSTM-CRF [7], Bi-LSTM-ATT [18] and Highway-BiLSTM [19]. Each of them is introduced as follows:

The CRF is proposed by Ren et al. [16] to recognize predicates, where manually designed rules are used to guarantee the uniqueness of predicates. In this experiment, we directly implement this model on the evaluation data set.

The LGN is a graph neural network model [26], where lexicon knowledge is used to capture local and global semantic dependencies in a sentence. This model was designed to support named entity recognition. In this paper, it is implemented to support predicate recognition.

The Bi-LSTM-CRF is a typical sequence labelling model [6], where a Bi-LSTM layer is adopted to capture previous and backward semantic dependency. Then, a CRF layer is stacked for constraining the structure of labelling sequence, where the dependency between adjacent labels is guaranteed.

The Bi-LSTM-ATT is a deep architecture proposed by Li et al. [18], where an attention-BiLSTM-CRF model is used to extract possible predicates from a sentence. Then, to satisfy the uniqueness of predicates, a convolutional neural network is adopted to select the most likely predicate.

The Highway-BiLSTM is a model proposed by Huang et al. [19], which stacks a multi-layer BiLSTM network and a Highway network for sequence labelling. The output path was optimized by a constraint layer which was designed to reduce the influence of semantic vanishing problem.

In the above five models, LGN and Bi-LSTM-CRF were originally designed to supported named entity recognition. The other three models were focusing on predicate recognition. In this experiment, they are implemented with the same data and settings for making a comparison. The result is shown in Table 2.

Table 2. Experimental results

Model	P	R	F
CRF [15]	75.26	58.12	65.59
LGN [25]	73.45	71.12	72.27
Bi-LSTM-CRF [6]	76.05	71.45	73.68
Bi-LSTM-ATT [17]	76.60	74.48	76.58
Highway-BiLSTM [18]	80.75	80.09	80.42
Ours	**79.99**	**81.60**	**80.78**

The CRF is a shallow architecture based on manually designed features. Because only lexical features are employed to make a prediction, it suffers from the feature sparsity problem caused by the fact that a sentence contains only a limited number of words. The LGN is a graph neural network model. It can encode semantic information from word embeddings initialized from external resources. Furthermore, it contains a graph neural network, which can learn structure information of a sentence. Therefore, the performance is improved considerably. Comparing with LGN, the advantage of Bi-LSTM-CRF is that a CRF is used to constrain the structure of labelling sequence. It is helpful for predicate recognition. The Bi-LSTM-ATT adopts a convolutional neural network to select the most likely predicate, which further improve the performance. The Highway-BiLSTM is effective to encode semantic information of a sentence. It is also effective to reduce the influence of semantic vanishing problem. Therefore, it achieved higher performance.

Comparing with related works, our bounding box based predicate recognition achieves competitive performance. It has three advantages to support predicate recognition. First, feature maps are abstract representations of a sentence, where semantic dependencies between words are encoded. Therefore, bounding boxes are abstract representations of possible predicate candidates, where global information is included. Second, a regression layer is adopted to locate predicates in a sentence. Because the regression task and the classification task share the same parameters in the bottom of the neural network, it is helpful to learn semantic information from a sentence. Third, it is an end-to-end model in which the predicate with the highest confidence score is selected as the output of this model. This strategy can guarantee the uniqueness of predicate in a sentence. Furthermore, an end-to-end model has a unique optimization objective. It can avoid the cascading failure caused in pipeline framework.

4.4 Influence of non-maximum suppression algorithm

In the regression process, location offsets of bounding boxes are learned, which leads to that many bounding boxes will be stacked in the neighborhood of a possible predicate. In order to remove redundant borders in the testing stage, non-maximum suppression algorithm is implemented to select the bounding box with the highest confidence score. As discussed in Section 3.3, in this paper, a 1D-NMS algorithm is used to select bounding boxes that are most likely true, where a threshold is adopted bounding boxes that have higher overlapping rates ($\lambda \geq 0.65$). Figure 2 shows the influence of on-maximum suppression on the final performance.

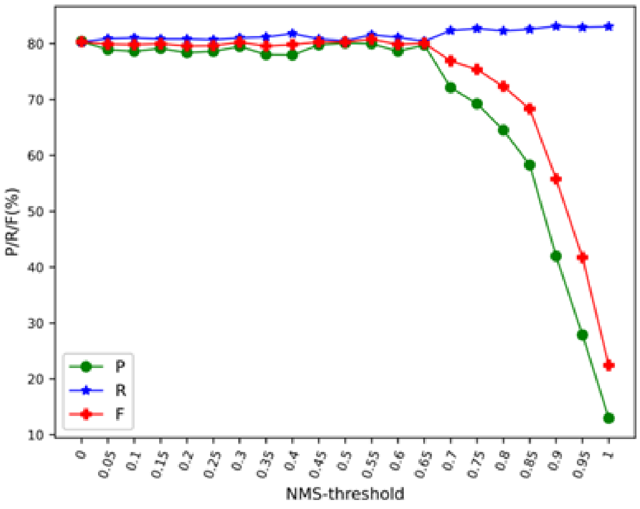
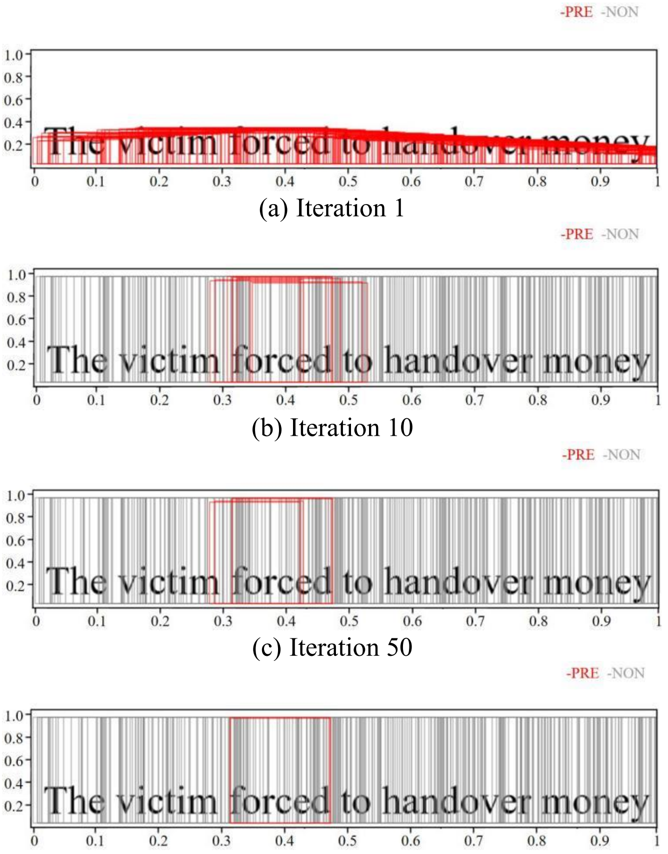


Figure 2. Influence of non-maximum suppression

The result shows that NMS are influential, especially when the NMS threshold is large than 0.65. The NMS collects a bounding box with the highest confidence score as reference. Then, comparing with the reference box, bounding boxes with overlapping ratio larger than 0.65 are discarded. Because predicate recognition has two class categories and a sentence only has a predicate, recognized predicates are close with each other. When the NMS threshold is increasing from 0 to 0.65, discarding bounding boxes with a lower overlapping ration can filter many noise boxes. However, when the NMS is larger than 0.65, only high overlapping bounding boxes are discarded. Many surrounding boxes are remained, which leads to a higher recall, but worsen the precision seriously.

4.5 Visualisation of bounding box regressing

In traditional models, positions of linguistic units are represented as discrete values. Recognizing linguistic units is modeled as a classification task, which predicts a class label based on features about a linguistic. In bounding box-based model, positions of predicates are represented as continuous values. A regression operation is used to predict the position offset of the bounding box relative to a true bounding box. The regression has been widely used to support object detection in the field of image processing [27-29]. However, it is rarely used to support linguistic unit recognition. In order to show the effectiveness of bounding box regression, in this section, the regression process is visualized to prove the feasibility of locating the predicate in the sentence.



(a) Iteration 1

(b) Iteration 10

(c) Iteration 50

(d) Iteration 100

Figure 3. Visualisation of bounding box regressing

In the testing phase, we selected a sentence from the test set. Then, in the training process, we recognize predicates from the selected sentence when the model was trained after n iterations ($n = \{1,10,50,100\}$). The outputs are visualized to show the process of bounding box regression. The result is shown in Figure 3.

As shown in Figure 3, the horizontal coordinates represent the length of the sentence. It is map into the range [0, 1]. The high of bounding boxes represents the classification confidence of the predicate. From Figure 2(a) to Figure 2(d), when the number of iterations increases, the bounding box has two trends. First, the classification confidence of the bounding box is increased. Second, the position of the bounding box is close to the real predicate. The result shows that the regression operation is feasible to locate predicate in a sentence.

5. CONCLUSION

In this paper, a boundary regression based model is proposed to support Chinese predicate recognition. It is an end-to-end multi-objective learning model, which simultaneously predicts the class probability and locates predicate in a sentence. Because bounding boxes are generated from feature maps transformed by Transformer, they encode semantic dependency information about a sentence. Experiments show that it is effective to support predicate recognition. Furthermore, the visualisation shows that boundary regression is impressive to support linguistic unit recognition. In further work, the regression operation can be extended to support other NLP tasks.

ACKNOWLEDGMENT

This work is supported by National Natural Science Foundation of China (Grant No.: 62166007).

REFERENCES

[1] Zhang, Y.S., Huang, G.J. (2012). Design and implementation of full-text retrieval system for people's daily annotated corpus. In Applied Mechanics and Materials, Trans Tech Publications Ltd., 135: 369-374. https://doi.org/10.4028/www.scientific.net/AMM.135-136.369

[2] Rabiner, L., Juang, B. (1986). An introduction to hidden Markov models. Ieee Assp Magazine, 3(1): 4-16. https://doi.org/10.1109/MASSP.1986.1165342

[3] Wu, G.D., Tang, H.D., Deng, Y.C., Wu, H.Q., Lin, C.R. (2022). A Data Driven Approach to Measure Evolution Trends of City Information Modeling. Journal of Urban Development and Management, 1(1): 2-16. https://doi.org/10.56578/judm010102.

[4] Lafferty, J., McCallum, A., Pereira, F.C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. Penn Engineering.

[5] Hammerton, J. (2003). Named entity recognition with long short-term memory. In Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003, 172-175.

[6] Huang, Z., Xu, W., Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. Computation and Language. https://doi.org/10.48550/arXiv.1508.01991

[7] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J.,

Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems 30.

[8] Luo, Z.S., Zheng, B.X. (1994). An approach to the automatic analysis and frequence statistics of Chinese sentence patterns. Journal of Chinese Information Processing, 8(2): 1-19.

[9] Li, G., Meng, J. (2005). A method of identifying the predicate head based on the correpondence between the subject and the predicate. Journal of Chinese Information Processing, 19(1): 1-7.

[10] Chen, X.H, Shi, D.X. (1997). Topic-subject labeling of Chinese sentences. National Computer Linguistics Joint Academic Conference, Beijing: China Computer Federation, 102-108.

[11] Wang, H., Zhou, G. (2010). Feature engineering for predicate identification and classification in semantic analysis. Computer Engineering and Application, 46(9): 134-137. https://doi.org/10.3778/j.issn.1002-8331.2010.09.038

[12] Chen, Z.Q. (2007). Study on recognizing predicate of Chinese sentences. Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications), 42(17): 176-178. https://doi.org/10.3321/j.issn:1002-8331.2007.17.055

[13] Han, L., Luo, S.L., Pan, L.M., Wei, C. (2014). High accuracy Chinese predicate recognition method combining lexical and syntactic feature. Journal of ZheJiang University (Engineering Science), 48(12): 2107-2114. https://doi.org/10.3785/j.issn.1008-973X.2014.12.002

[14] Gong, X., Luo, Z., Luo, W. (2003). Recognizing the predicate head of Chinese sentences. Journal of Chinese Information Processing, 17(2): 7-13. https://doi.org/10.3969/j.issn.1003-0077.2003.02.002

[15] Ren, X., Zhou, Q., Kit, C., Cai, D. (2010). Automatic identification of predicate heads in Chinese sentences. In CIPS-SIGHAN Joint Conference on Chinese Language Processing.

[16] Li, L., Zhao, W.N., Zewang, K.Z. (2019). Tibetan predicate verb phrase recognition model based on word vector features Electronic Technology and Software Engineering, 4: 242-243

[17] Li, T., Qin, Y.B., Huang, R.Z., Cheng, X.Y., Chen, Y.P. (2020). Research on Chinese predicate verb recognition based on neural network. Journal of Data Acquisition and Processing, 35(3): 582-590. https://doi.org/10.16337/j.1004-9037.2020.03.020

[18] Huang, R.Z., Jin, W.F., Chen, Y.P., Qin, Y.B., Zheng, Q.H. (2021). Research on Chinese Predicate Head Recognition Based on Highway BiLSTM Network Journal of Communication, 1: 100-107. https://doi.org/10.11959/j.issn.1000-436x.2021027

[19] Chen, Y., Zheng, Q., Chen, P. (2015). A boundary assembling method for Chinese entity-mention recognition. IEEE Intelligent Systems, 30(6): 50-58. https://doi.org/10.1109/MIS.2015.71

[20] Wang, J., Shou, L., Chen, K., Chen, G. (2020). Pyramid: A layered model for nested named entity recognition. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 5918-5928. https://doi.org/10.18653/v1/2020.acl-main.525

[21] Yang, S., Tu, K. (2021). Bottom-up constituency parsing and nested named entity recognition with pointer networks. arXiv preprint arXiv:2110.05419. https://doi.org/10.48550/arXiv.2110.05419

[22] Shen, Y., Ma, X., Tan, Z., Zhang, S., Wang, W., Lu, W. (2021). Locate and label: A two-stage identifier for nested named entity recognition. arXiv preprint arXiv:2105.06804. https://doi.org/10.48550/arXiv.2105.06804

[23] Chen, Y., Wu, L., Zheng, Q., Huang, R., Liu, J., Deng, L., Chen, P. (2022). A boundary regression model for nested named entity recognition. Cognitive Computation, 1-18. https://doi.org/10.1007/s12559-022-10058-8

[24] Devlin, J., Chang, M.W., Lee, K., Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805. https://doi.org/10.48550/arXiv.1810.04805

[25] Chen, Y., Jin, W., Qin, Y., Huang, R., Zheng, Q., Chen, P. (2021). Annotation of chinese predicate heads and relevant elements. arXiv preprint arXiv:2103.12280. https://doi.org/10.48550/arXiv.2103.12280

[26] Gui, T., Zou, Y., Zhang, Q., Peng, M., Fu, J., Wei, Z., Huang, X.J. (2019). A lexicon-based graph neural network for Chinese NER. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pp. 1040-1050. https://doi.org/10.18653/v1/D19-1096

[27] Erhan, D., Szegedy, C., Toshev, A., Anguelov, D. (2014). Scalable object detection using deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 2147-2154.

[28] Redmon, J., Divvala, S., Girshick, R., Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788.

[29] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C. (2016). Ssd: Single shot multibox detector. In European conference on computer vision, Springer, Cham, 9905: 21-37. https://doi.org/10.1007/978-3-319-46448-0_2