[1]Jiang Tao Zheng

Research on the discrimination of translation difficulty level based on spoken language signal processing technology

Abstract: - Computer translation systems require the ability to use external dictionaries in order to quickly improve the accuracy of professional field translations, and establishing a more accurate grading and evaluation system has important research value. This article selects spoken signals as the pre training processing object, establishes a pre training ELMo model based on optimized similar word vectors based on spoken signals, and applies this model and method to Chinese logistics outbound call data. Subsequently, compared with the traditional industry best translation mechanism, this paper proposes an improved machine translation intervention method based on dictionary guided decoding. This method, given a Transformer baseline model, supervises additional attention heads during the training process, which can achieve better intervention success rate and translation quality after intervention. Finally, using a tree shaped multi-layer combination model, a tree shaped two-layer combination model is constructed. Several BP neural networks are used as the lowest level classifiers, and the LVQ neural network is used as the upper level classification combiner to train the DuIE2.0 dataset. Experimental results have shown that compared with the BERT multi head model, the F1 value of the proposed model in this chapter has increased by 1.85%.

Keywords: Spoken signals; Multi-layer feedforward network; Difficulty grading; RoBERTa; Pre-training

1. Introduction

Eliminating language barriers between people has been a human endeavor since ancient times, and the Polish-Jewish Dr. Zaimenhof founded Esperanto on the basis of Indo-European language in 1887 to eliminate language barriers in international communication. However, with more than 1,900 languages in use in the world today, people from different countries who want to communicate well may need to spend a significant portion of their time and energy learning a new language[1,2]. Some experts have even pointed out that the language barrier has become a major constraint to the globalization of society in the 21st century. Fortunately, with the rapid development and popularization of communication technology, the dramatic increase in the amount of information, the increasingly rich means of communication, and the continuous progress of science and technology. It is no longer a dream to use machines to realize translation between different languages.

Now, with the frequent utility of the Internet and the growing frequency of global social exchanges, usual human translation has been a long way from being in a position to meet the unexpectedly developing demand for

[1] Henan Vocational College of Information and Statistics School of Public Foundation,Graduated from: Henan University,School of Foreign Language, China

E-mail: can19830817@sina.com

translation, in particular in the spoken language, and human beings are eagerly searching ahead to a extra smart spoken language translation. This mission starts offevolved from desktop translation of spoken language, combining the traits of spoken language itself, with the intention of exploring the desktop translation issues precipitated by using the structural variations between one-of-a-kind languages. Taking the usual desktop translation machine MOSES as a platform, including the exploration of null lessons and zero pronouns, and aiming at enhancing the impact of spoken translation, the corpus in the area of Chinese-English dialogues is studied and analyzed. And attempts were made from multiple angles[3].

With the continuous innovation of deep learning technology, from template matching to machine learning algorithms to the technical precipitation of neural networks, various models of deep learning have good application prospects in computer vision, machine translation and other aspects. Deep learning models also bring new opportunities for entity-relationship extraction so that researchers do not have to spend a lot of time on feature engineering and can focus on fine-tuning the model parameters to improve the model effect. In this paper, we focus on the Chinese entity-relationship extraction task under the deep learning model, which utilizes a pre-trained language model and incorporates the learning method of knowledge representation for relationship extraction, so as to improve the performance of the proposed model in the Chinese entity-relationship extraction task.

2. Speech signal recognition based on enhanced pre training

Enhancing the illustration potential of phrase vectors of pre-trained language fashions is one of the warm lookup instructions in the cutting-edge lookup on the robustness of spoken intention recognition. In order to decorate the overall performance of the pre-trained language mannequin in the spoken intention awareness task, it is crucial to mix the beneficial records output from the speech consciousness module and follow it fairly to the coaching over system of the pre-trained language model. In this chapter, the pre-trained ELMo model based on optimized similar word vectors is replicated on logistics outbound call data, and the experiments compare and analyze the effects of two feature extraction methods, average pooling and maximum pooling, on the experimental results[4]. On the Bert pre-training task, the idea of optimizing similar word vectors is extended to incorporate phoneme information into the Bert pre-training language model task, so that the word vector representations have phoneme feature information, which enhances the robustness of the word vector representations of the Bert pre-training model under noisy text.

2.1 Pre-trained language model with optimized similar word vectors

Taking ELMo, a publicly available pre-trained language model from Allennlp's lab, as an example, the pre-trained language model employs a character-level convolutional neural network (Character CNN) as the initial input layer feature extraction module, followed by a two-layer bi-directional long- and short-term memory network (Bi-LSTM) coupled with an autoregressive language model to model the contextual relationships of the words in a sentence.

Given a sentence $s = \{w_1, w_2, \ldots, w_n\}$, the loss of the autoregressive language model modeled by the bidirectional long short-term memory network is calculated using the following formula:

$$L_{LM} = \frac{1}{n} \sum_{t=1}^{n} -\log\, p(w_t \mid w_{<t}) - \log\, p(w_t \mid w_{>t}) \tag{2.1}$$

The two equations in Eq. represent the probability calculation of forward language model and reverse language model, respectively. However, such a traditional way of modeling pre-trained language models does not achieve the purpose of bringing the embedding representations of similar words closer together. When measuring the similarity of two words, the cosine distance is generally used to calculate the similarity size of two word vectors, and the cosine similarity calculation formula is as follows:

$$similarity = cos(\theta) = \frac{A \cdot B}{||A|| \cdot ||B||} \tag{2.2}$$

For the collected speech-recognized text and manually labeled text, the experiment adopts $c = \{w_{t1}^{trs}, w_{t2}^{asr}\}$ as the confusing pair pairs of two words in corresponding positions in a training sample, where trs and asr denote the manually labeled text and the speech-recognized text, respectively, in a training sample, and the set of easy-to-confuse pairs is denoted by $C = \{c_1, c_2, c_3, \ldots, c_n\}$. then the calculation of the LOSS for the two corresponding positions of the similar words in the language model in a parallel training sample can be calculated by the following formula.

$$L_{\cos} = \frac{1}{|C|} \sum_{c \in C} \sum_{i=n}^{1} 1 - \frac{A \cdot B}{||A|| \cdot ||B||} \tag{2.3}$$

2.2 Model Architecture

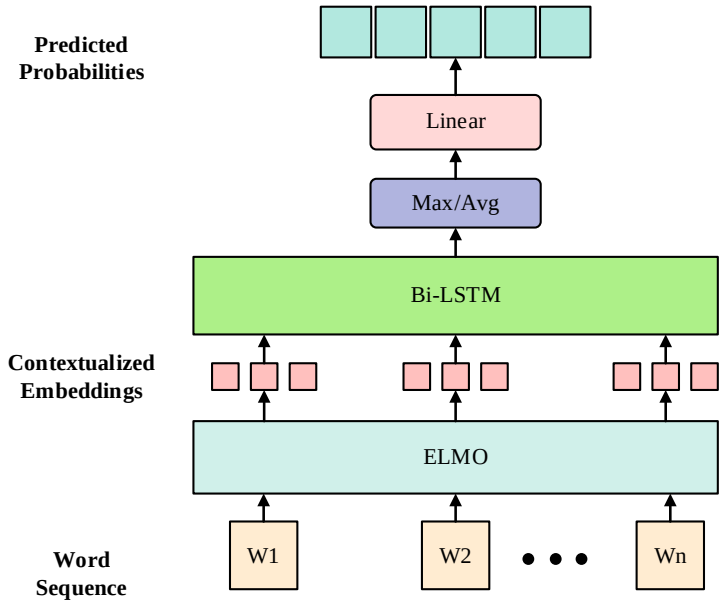

Figure 2.1 Model structure diagram

2.3 Bi-LSTM layer

The output vectors of the hidden layer of the fine-tuned pre-trained language mannequin are used as the enter vectors of the intent attention model. Subsequently, the enter vectors are changed the usage of the Bi-LSTM model, and then the output effects are processed the use of the two strategies of most pooling and common pooling methods, and in the end the received phrase vectors will be accessed to the fully-connected layer for function combination. And then finally scaled by the softmax function to obtain the category with the largest probability

of the corresponding class as the prediction result.The output vector of the forward hidden layer of Bi-LSTM model can be expressed as $\{\vec{h_1}, \vec{h_2}, \vec{h_3}, \dots, \vec{h_n}\}$, and the reverse hidden layer output vector can be expressed as $\{\overleftarrow{h_1}, \overleftarrow{h_2}, \overleftarrow{h_3}, \dots, \overleftarrow{h_n}\}$. The final output representation vector $\{h_1, h_2, h_3, \dots, h_n\}$ of the Bi-LSTM layer is obtained by splicing the output vector of the forward hidden layer and the output vector of the reverse hidden layer, where $h_i = [\vec{h_l}, \overleftarrow{h_l}]$.

2.4 Experimental setup and result analysis

Table 2.1. Parameterization of the model

hyper parameterization	parameterization
ELMo embedding size	1024
Batch size	128
Bi-LSTM hidden	400
Bi-LSTM layer	3
Full Connection Hidden	64
learning rate	0.001
Iteration epoch	10
Dropout Ratio	0.3

In terms of pre-training language models, the original open-source ELMo model of the Allennlp Institute, the fine-tuned ELMO model using the Snips dataset(as shown in Table 2.1), the ELMo model with word vector cosine similarity fine-tuning using supervised data, and the ELMo model with word vector cosine similarity fine-tuning using unsupervised data were used for the experiments as follows: using f-ELMo to denote the ELMo model with ordinary fine-tuning using Snips data, and sup-ELMo and unsup-ELMo to denote the ELMo model after fine-tuning using supervised data and the ELMo model after fine-tuning using unsupervised data, respectively[5,6]. The experimental test set uses speech recognition text (ASR) data and real manually labeled data (Trs) on the Snips dataset. The experimental results are shown in Table 2.2.

Table 2.2 Snips data experiment results

Model	Pooling	Trs acc(%)	ASR acc(%)
ELMo	avg	96.44	80.41
ELMo	max	94.51	80.89
ft-ELMo	avg	97.26	85.41
ft-ELMo	max	96.38	87.65
sup-ELMo	avg	95.57	87.91
sup-ELMo	max	95.77	88.31
unsup-ELMo	avg	95.09	90.54
unsup-ELMo	max	95.22	90.07

It can be observed through the experimental results in Table 2.2 that the experimental prediction accuracy results

using only the Allennlp open source ELMo pre-trained model on Snips' Trs test set are better, with the corresponding prediction accuracies obtained by the average pooling approach and the maximum pooling approach being 96.44% and 94.51%, respectively, but the results on the ASR test dataset show poorer performance with the corresponding The prediction accuracies obtained by the average pooling approach and the maximum pooling approach are 80.41% and 80.89%, respectively, which illustrates that the traditional pre-trained language model is not well adapted to the task of recognizing spoken intention corresponding to speech recognition text containing noise. When the Snips data is used to fine-tune the ELMo model, the corresponding experimental results on the ASR test set have a large improvement, compared to the original pre-trained model without fine-tuning has an accuracy improvement of about 6% on the ASR test set, of which the use of average pooling has an accuracy improvement of 5.03% on the ASR dataset, and the use of maximum pooling has an accuracy improvement of 6.67%. accuracy improvement, the experimental results demonstrate the effectiveness of fine-tuning language models using domain data.

2.4 Phonetic Bert Model

The NSP task mainly models the sequential relationship between sentences, which enables the Bert model to not only model the contextual relationship in a sentence, but also capture the correlation between sentences, so that the Bert model can be used at the chapter level to model the contextual relationship of a sentence, and the Bert model is not only a model of the contextual relationship, but also an interlinked model. Bert model to perform well in chapter-level tasks as well. Bert originally used 15% of the words in a sentence to be masked, and in 80% of the cases of the 15% of the words that were masked the words were replaced using the special token [MASK], in 10% of the cases random substitutions occurred, and in 10% of the cases the original words were left unchanged. Because of the need to add phoneme information to the language model, some mechanisms of MLM need to be modified when modeling the language model. Since the phoneme sequence and the sentence sequence can be regarded as two parallel sentences as inputs, there is a complementary relationship between the amount of information in the inputs, so it is necessary to increase the proportion of masked words, and the proportion of sentence masking is set to 20% in the experiment.

In order to enhance the correlation between Chinese characters and phonemes, two masking methods are used for the input training data[7,8]. The first way is to mask both the word and the corresponding phoneme at the corresponding position, and predict the word and phoneme at the same position through the context The second way is to mask the word and the corresponding phoneme at the corresponding position separately, and predict the word and the corresponding phoneme at the masked position through the context and the corresponding phoneme and word. The mixture of the two approaches enhances the connection between word and phoneme and context. The two masking approaches combining phoneme information are shown in Figure 2.2 and Figure 2.3.

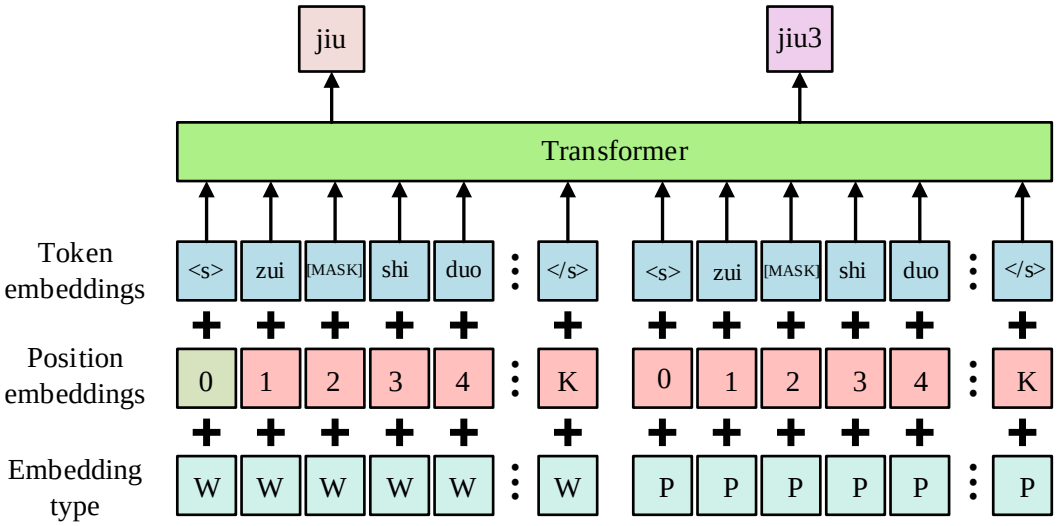

Figure 2.2 Bert Mask Approach I

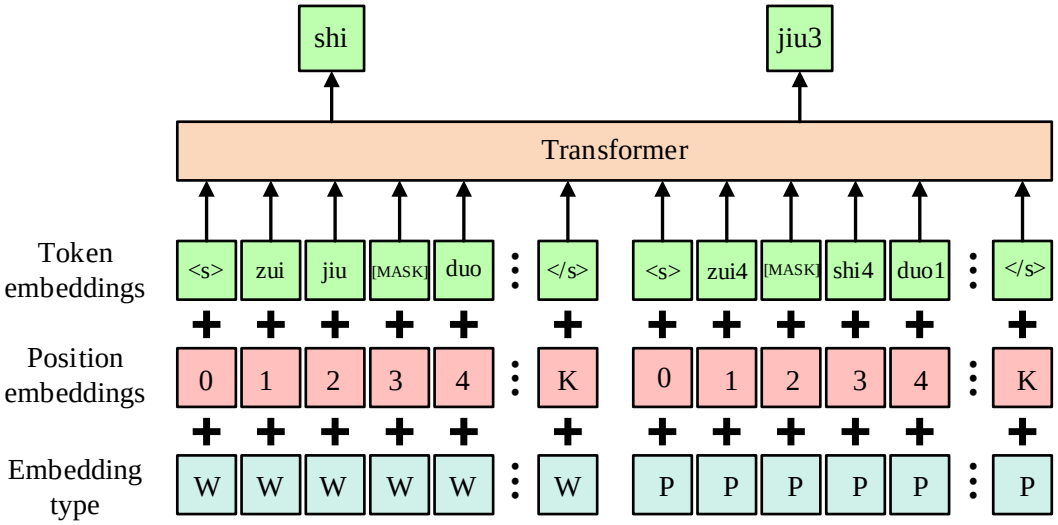

Figure 2.3 Bert Mask Approach II

3. Intervention method based on oral signal alignment optimization and decoding constraints

The methods proposed in this chapter directly utilize intervening translations specified in advance to guide the decoding process.Inspired by the work of Strubcll et al. who utilized external supervisory signals containing syntactic information to enhance the effectiveness of Transformer, the methods proposed in this chapter utilize a dedicated attention head to learn word alignment from external word alignment signals[9,10]. Established methods usually extract word alignment information from the weights generated by Transformer's generalized attention head, distinguishing from established methods, the method proposed in this chapter utilizes a dedicated attention head to learn the exact word alignment and use it to guide the decoding process for neural network machine translation intervention.

3.1 Extracting word alignment from the original Transformer

A common and simple way to obtain word alignment from a Transformer is to select the source word that has the largest of the attentional weights of the current target end word over all source words. More specifically, the source end word aligned to the current target end word is determined by selecting the source end position with the largest cumulative attentional weight of.

$$\gamma(t) = \underset{i\in\{1,\cdots,m\}}{\operatorname{argmax}} \frac{1}{N}\sum_{j=1}^{N} \alpha_{t,i}^{j} \tag{3.1}$$

Where *i* is a candidate source-side word aligned with the current word at the target side. At step *t* of the decoding process, for the jth layer network at the target end, $\alpha_{t,i}^{j}$ is the attentional weight of the ith position in the source end sentence.

3.2 Supervised learning of word alignment using a dedicated attention head

Inspired by the work of Strubell et al, the work in this chapter extends the architecture of Transformer's original multi-head self-attention mechanism by adding an additional attention head that is supervised trained with external word alignment information. As shown in Figure 3.1, an additional attention head is added to the target-side-to-source-attention subnetwork layer in each network layer of the decoder.

More specifically, at layer *j* of the decoder, at step *t* of the decoding process, after computing the hidden state $s'_{t,j}$ of the first self-attention sub-network layer, two types of attention weights are computed in the target-to-source attention sub-network layer: the original multiple-head attention weights and the additional attention weights. Both attentional weights are computed using the same query (Q), key (K), and value (V), but are used for different purposes:.

(1) The original multi-head attention produces weights $\{\alpha_{t,1}^{j}, \alpha_{t,2}^{j}, \cdots, \alpha_{t,m}^{j}\}$ for all source-end positions $\{1, \cdots, m\}$ is used to produce the candidate hidden state $s''_{t,j}$, which is what the standard Transformer does.

(2) The additional attention head generates intentional weights $\{\beta_{t,1}^{j}, \beta_{t,2}^{j}, \cdots, \beta_{t,m}^{j}\}$ which are used for end-to-source word alignment, because this attention head is utilizing the external word alignment information to do the supervised learning during the training process.

In order to be able to supervise this additional attention head using external word alignment information, the methods in this chapter introduce a word alignment loss function:

$$L_t^{\text{align}} = -\log \sum_{i=1}^{m} (\overline{\beta}_{t,i} \times \hat{\alpha}_{t,i}) \tag{3.2}$$

Where $\overline{\beta}_{t,i}$ is the average of the attentional weights of all network layers of the decoder to the ith position at the source:

$$\overline{\beta}_{t,i} = \frac{1}{N}\sum_{j=1}^{N} \beta_{t,i}^{j} \tag{3.3}$$

Moreover, according to the external word alignment supervision information, $\hat{\alpha}_{t,i}$ is set to 1 only when the target-side word y_t is aligned with the source-side word x_i, and 0 otherwise. The final objective function L consists of

two parts: the ordinary translation loss and the word alignment loss:

$$L = \sum_{t=1}^{n} (L_t^{lexical} + \lambda * L_t^{align}) \tag{3.4}$$

Where λ is empirically set to 0.3 and $L_t^{lexical}$ is the loss predicted by the original word:

$$L_t^{lexical} = -log\ (p(y_t \mid y_1, \cdots, y_{t-1}, \mathrm{x})) \tag{3.5}$$

3.3 Word Alignment Enhanced Transformer Model for Spoken Word Extraction

Unlike the basic word alignment extraction method described in Section 3.1, the method in this chapter uses only this additional specialized attention header described above to extract word alignments, and at step t of the decoding process, the source-end word aligned to the current translated word is computed by the following formula.

$$\gamma(t) = \underset{i\in\{1,\cdots,m\}}{\operatorname{argmax}} \overline{\beta}_{t,i} \tag{3.6}$$

Where $\overline{\beta}_{t,i}$ is the average of the attentional weights of all subnetwork layers of the decoder up to the ith position at the source end.

Unlike the work of Alkhouli et al. and Zenkel et al. the word alignment in the methods of this chapter is obtained by selecting the source location with the largest average attentional weight[11,12], which is generated by dedicated attention heads trained with supervised information from external word alignment information, rather than from the multiple attention heads that were the default for the original Transformer.

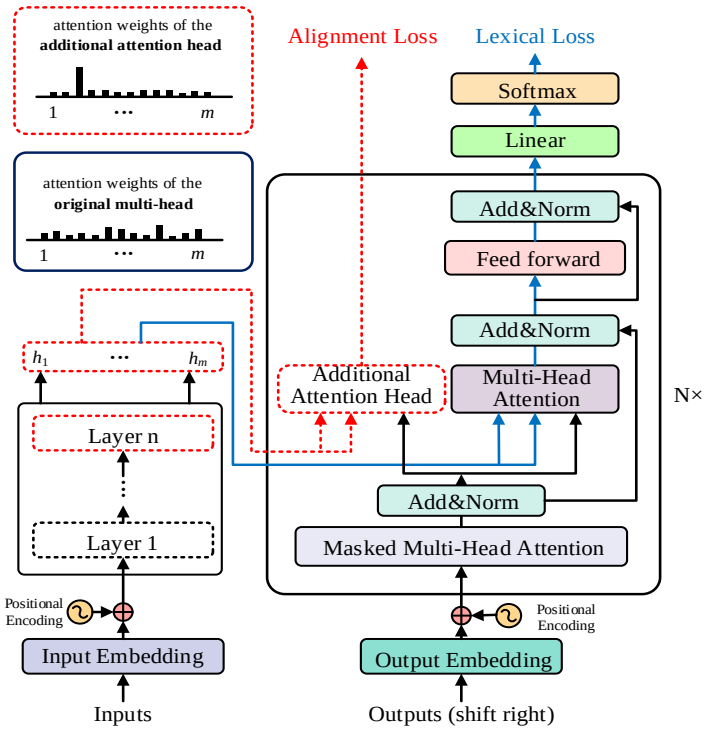

Figure 3.1 Adding an extra attention header to the original Transformer

3.4 Lexicon-guided decoding

In their 2018 work, Alkhouli et al. describe a "lexicon-based guidance of the decoding process" task, which is proposed as a downstream task that utilizes Transformer word alignment[13,14]. It provides an efficient way to guide the decoding process using pre-specified translations. More specifically, at step t of the decoding process, if the source word x_j aligned to the current translated word hits the lexicon that serves as the translation constraint (or "translation pre"), the probability distribution on the target word list output by the decoder is reset. The translation loss of all words in the target word list, except for the pre-specified target translations, is set to infinity, i.e., the predicted probability is close to zero.

Terminology Constraint: $X^{u:v} \rightarrow \{y_{c_1}, \ldots, y_{c_k}\}(u \leq j \leq v)$

Decoding steps:

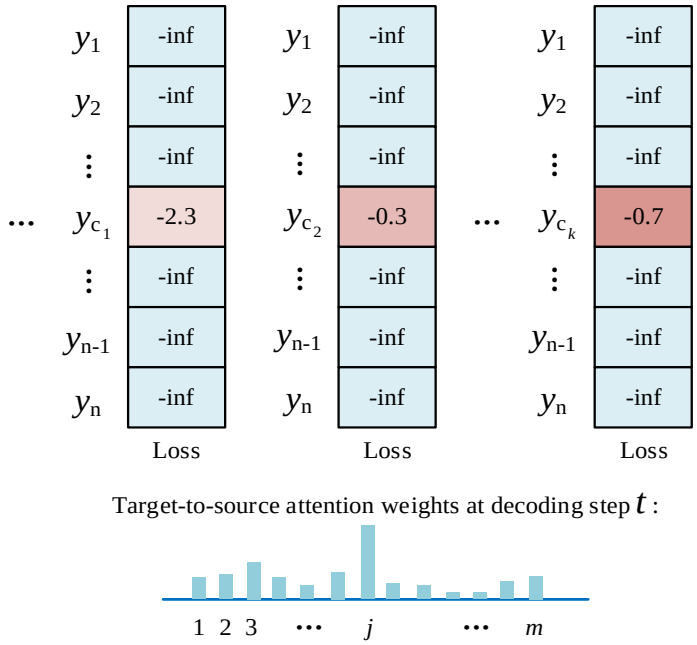

Figure 3.3 Schematic diagram of the decoding process using a dictionary to guide the decoding process

Matches when a pre-specified translation is made:

$$x^{u:v} \rightarrow \{y_{c_1}, \cdots, y_{c_k}\} \tag{3.7}$$

Where $x^{u:v}$ denotes a word or sub-word in the source end sentence from position u to position v, and $\{y_{c_1}, \cdots, y_{c_k}\}$ is an intervening translation specified in advance or provided by the user, which consists of *k* words (or sub-words) in the target vocabulary[15]. At the tth step of the decoding step, i.e., when generating the tth word (or subword) of the target side, if the jth position of the source side aligned with it is within (u,v), all the subsequent k decoding steps will be constrained.

4. Optimization Training of Difficulty Grading Based on Digital Signal Processing

The relationship classification subtask is usually performed after the entity recognition task is completed, which classifies the entity pairs after entity recognition into relationships and extracts the desired triples from them.

Compared with template-based methods, this method has better generalization ability, and does not require the design of rules, but only needs to select useful text features by human beings. giuliano et al. achieved the best results in the relational extraction task by selecting features such as lexical and syntactic information of entity contexts, and using the support vector machine algorithm. However, this learning method requires manual extraction of features, and the generalization ability and practicality are still relatively weak. With the massive wide variety of purposes of deep studying in herbal language processing tasks, Liu et al. proposed for the first time the use of convolutional neural networks (CNNs) to resolve the relationship extraction problem. Zeng et al. proposed a segmented convolutional neural community (Piecewise Convolutions Neural Network (PCNN)), based totally on the entity area information, to lift out the three pooling operations. can extensively enhance the relationship extraction overall performance of the model[16,17].

On this basis, this chapter proposes a new segmented convolutional neural network model named RSA-PCNN (RoBERTa and Self-Attention Piecewise Convolutions Ncural Network). We add entity information to the sentence input of the overall model, and use different convolution methods for sentence and entity information segmentation to fully utilize the existing entity information[18]. The RoBERTa pre-trained model is selected for vector representation, and a self-attention mechanism is introduced to give a higher proportion of weights to vectors containing correct relations and reduce error transmission, thus improving the extraction performance of the model.

4.1 Overall structure of the model

First, in order to better understand the phenomenon of multiple meanings of words as well as contextual information, the RoBERTa language model is used for the vector representation of words at the word vector representation layer. Then, the vector representation obtained from the previous layer is used as an input to the BiLSTM network structure. Finally, in order to solve the entity nesting problem, label prediction using global pointers is chosen to obtain entity information for named entity recognition[19]. Figure 4.1 below shows the structure of the overall model for named entity recognition.

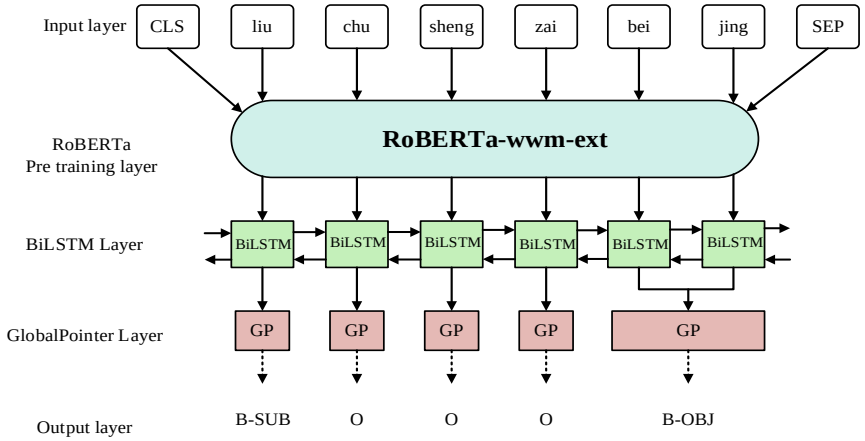

Figure 4.1 ucture of named entity recognition model

The positional embedding is calculated as shown in Equation (4.1) and Equation (4.2). Where pos represents the position where the current word is located and i represents the dimension size of the vector.

$$\mathrm{PE}(\mathrm{pos}, 2\mathrm{i}) = \sin\left(\mathrm{pos} / 10000^{2\mathrm{i}/\mathrm{d}_{\mathrm{model}}}\right) \quad (4.1)$$

$$\mathrm{PE}(\mathrm{pos}, 2\mathrm{i}+1) = \cos\left(\mathrm{pos} / 10000^{2\mathrm{i}/\mathrm{d}}\ \mathrm{model}\right) \quad (4.2)$$

Given an input sequence of sentences $S = \{s_1, s_2, s_3, \ldots, s_n\}$, where two entities in the sentence can be represented as $e_1(s_m)$ and $e_1(s_z)$. After the contextualization of the sentence input sequence by the RoBERTa-wwm-ext pre-trained model, the vector representation is obtained as shown in Eq. (4.3) as follows.

$$W = \{w_1, w_2, w_3, \ldots, w_n\} \quad (4.3)$$

In addition, the inclusion of position vectors has an improvement in entity recognition. The position vector $p_i(i = 1,2,3,\ldots,n)$ of each word is spliced by two vectors, one of which is the relative distance $d_{i1}(i = 1,2,3,\ldots,n)$ between the word and entity e_1, and the other is the relative distance $d_{i2}(i = 1,2,3,\ldots,n)$ between the word and entity e_2. ...,n). Finally, the word vectors and position vectors of each word are combined together to form a vector representation of the word.

In this paper, we choose to use the RoBERTa-wwm-ext pre-training model, which is an improved version of the BERT-wwm-ext model released by the joint lab of HIT and KUDA, to train the word vectors.The input representation of the RoBERTa model is shown in Figure 4.2. The RoBERTa model is essentially a tuned upgrade of BERT, and the new pre-training strategy is adopted to achieve better performance in downstream tasks. Since the RoBERTa model is proposed on the basis of English data, directly using the original RoBERTa model on Chinese dataset will directly affect the model effect. Therefore, for the Chinese dataset, the biggest difference between RoBERTa-wwm-ext proposed by Xunfei Joint Laboratory of HIT University and the original model is the selection of the whole word masking strategy in the pre-training stage considering the characteristics of the Chinese language[20,21]. In order to higher research the contextual information, the mannequin shape chooses the equal multilayer bi-directional Transformer as the encoder as the BERT model, and the shape of the Transformer mannequin is proven in Figure 4.2 below.

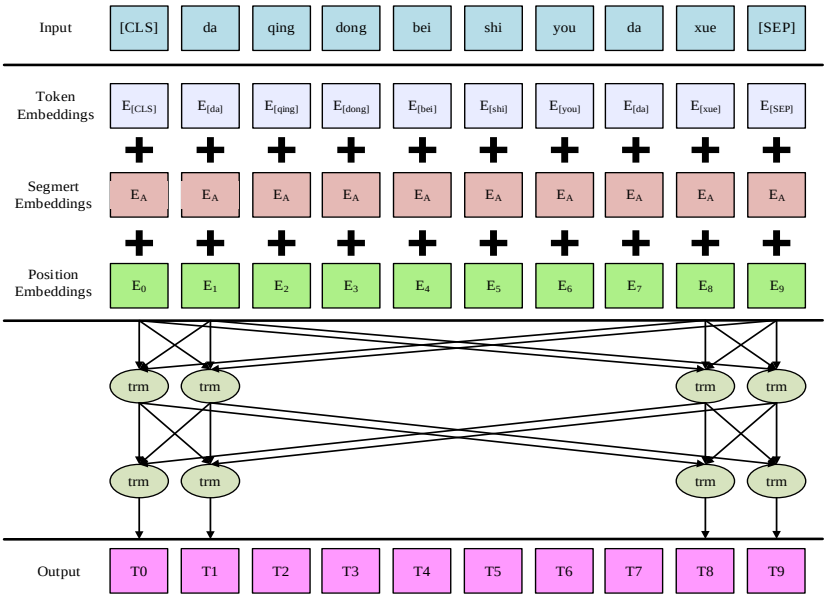

Figure 4.2 RoBERTa model input vector composition

Given the input data $X = (x_1, x_2, x_3, \dots, x_n)$, each gating and state is calculated as shown in the following equation.

$$i_t = \sigma(w_{ix}x_t + w_{ih}h_{t-1} + b_i) \quad (4.4)$$

$$f_t = \sigma(w_{fx}x_t + w_{fh}h_{t-1} + b_f) \quad (4.5)$$

$$C_t = f_t * C_{t-1} + i_t * Ct \quad (4.6)$$

$$o_t = \sigma(w_{ox}x_t + w_{oh}h_{t-1} + b_o) \quad (4.7)$$

Where *w* represents each weight matrix, w_{ix} is the weight matrix from the input gate to the output, *b* represents the bias vector whose subscripts correspond to the bias vectors of its different gates. σ is the sigmoid function, *i,f,o,C* represent the three control gates and the update state, respectively, C_t refers to the abstraction information, and * represents the dot product.

In the training process of Chinese entity recognition, suppose the length of the input sentence is *n*, after encoding, we get the vector sequence $h_t = [h_1, h_2, \dots, h_n]$, by transforming $q_{i,\alpha} = W_{q,\alpha}h_i + b_{q,\alpha}$ and $k_{i,\alpha} = W_{k,\alpha}h_i + b_{k,\alpha}$. After the transformation we get two vector sequences $[q_{1,\alpha}, q_{2,\alpha}, \dots, q_{n,\alpha}]$ and $[k_{1,\alpha}, k_{2,\alpha}, \dots, k_{n,\alpha}]$, which are also used to identify the vector sequences used for the type α entities. Here, we define an entity scoring function for α as shown in Eqs. (4.8).

$$s_\alpha(i,j) = q_{i,\alpha}^T k_{j,\alpha} \quad (4.8)$$

The inner product of $q_{i,\alpha}$ and $k_{i,\alpha}$ is used as the scoring of the segment $t_{[i:j]}$ which is an entity of type α. Here, $t_{[i:j]}$ refers to the consecutive substring composed of the i-th to j-th elements of the sequence *t*.

In order to better recognize the entity information, we add the relative position information and adopt the transformation matrix R_i in the relative position encoding of Transformer. applying it to g and k of the above equation yields the following equation (4.9) shown below.

$$s_\alpha(i,j) = (R_i q_{i,\alpha})^\top (R_j k_{j,\alpha}) = q_{i,\alpha}^\top R_i^\top R_j k_{j,\alpha} = q_{i,\alpha}^\top R_{j-i} \boldsymbol{k}_{j,\alpha} \quad (4.9)$$

The loss function uses a generalization of the single-objective multiclassification cross entropy function for multi-label classification, defined by the formula shown below:

$$\log\left(1 + \sum_{(i,j)\in P_\alpha} e^{-s_\alpha(i,j)}\right) + \log\left(1 + \sum_{(i,j)\in Q_\alpha} e^{s_\alpha(i,j)}\right) \quad (4.10)$$

Where P_α is the first and last set of all entities of type α for that sample, and Q_α is the first and last set of all non-entities or entities of non-type α for that sample, and satisfies the following requirement.

$$\Omega = \{(i,j) \mid 1 \le i \le j \le n\} \quad (4.11)$$

In the decoding phase, all segments $t_{[i:j]}$ satisfying $s_\alpha(i,j) > 0$ are considered as entity outputs of type α.

4.2 Segmented Convolutional Network Layers

Convolutional neural networks were originally applied to image processing tasks, Kim first used this network structure on a text categorization task. The advantage of convolutional neural networks of not having to extract

features manually and the ability to have a strong feature representation has led to their widespread use in relation extraction tasks[22]. The biggest change in the segmented convolutional neural network compared to the traditional convolutional neural network is that the operation of segmented maximum pooling is chosen at the end of the model, which is aimed at obtaining better structured information between two entities. Therefore, in this paper, we use segmented convolutional network for the relationship extraction task, and its structure is shown in Fig. 4.3 below, which is mainly divided into three network layers, among which the vector representation layer has been elaborated in the named entity recognition task, so we will not repeat it here. This section focuses on the segmented convolutional layer and pooling layer in detail[23].

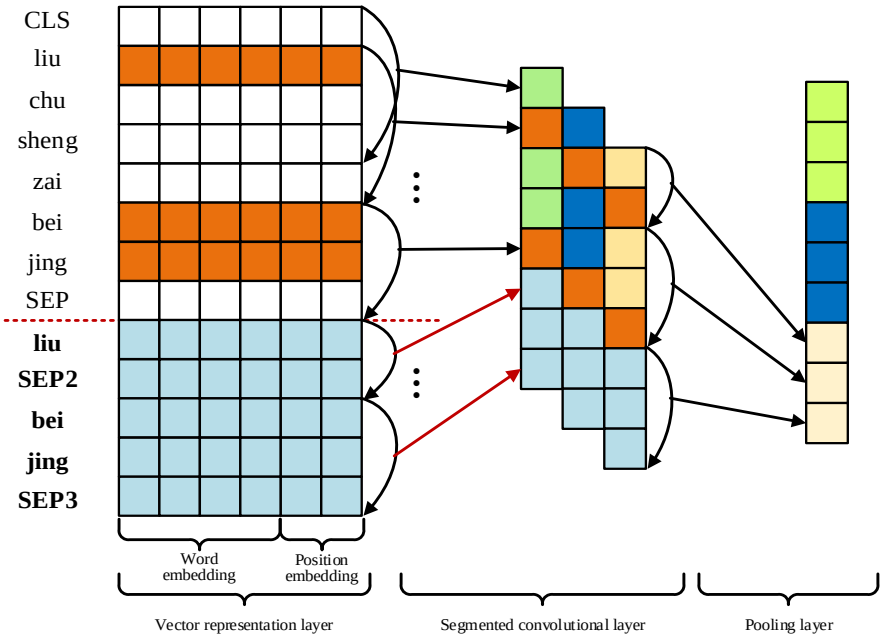

Figure 4.3 Segmented convolutional network model structure

The model integrates text with corresponding entity information into the input, resulting in a coded vector matrix. Based on the vector matrix, the segmented convolutional layer performs different convolutional operations on the sentence and entity information, respectively. We choose to perform the null convolution operation for sentences to obtain longer sentence text features without increasing the model parameters, and the ordinary convolution operation for entity information. Finally, the two convolutional operations are joined and input into the pooling layer.

Liu et al. first proposed the use of convolutional neural network model for relationship extraction task in 2013[24]. The main function of the convolutional neural network is to represent the important features in the sentence by sequentially extracting a specified number of words in the sentence as instances of the window through the convolutional kernel and outputting a multidimensional vector through the window[25]. In this paper, ordinary convolution operation is performed on entity information in segmented convolutional layer, and the convolutional layer performs convolution operation on the content of entity information as shown in Equation (4.12).

$$s(x) = \int_{-\infty}^{\infty} f(t)g(x-t)dt \tag{4.12}$$

Where f denotes the input function for the upper layer, g represents the kernel function, t denotes the

information about the current location, and *x* is the size of the radius.

For sentence information, we use the null convolution operation, which is also known as inflation or dilation convolution and is a variant of traditional convolution. It captures the long dependencies in the sentence, and hence, a wider range of valid input information can be obtained.In 2015 Yu et al. proposed the method of cavity convolution, which compares to the normal convolutional network and mainly changes the convolutional network by increasing the expansion rate factor on the normal convolutional network which has expansion rate factor 1[26,27]. The larger the dilatation rate, the larger the sampling range of the cavity convolution, and the fewer the sampling points in the relative unit range. Null convolution has the same number of convolution kernels as normal convolution, yet it can have a larger sensory field. Figure 4.4 shows the process of adding a hole to a 3x3 normal convolution to form a 3x3 cavity convolution with a dilation of 2.

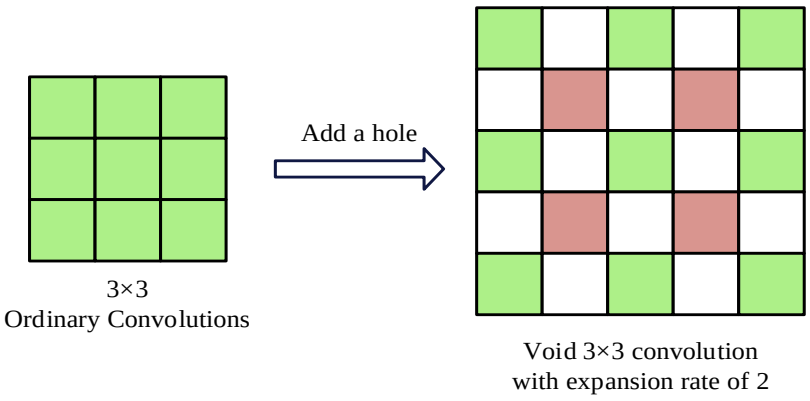

Figure 4.4 Transformation process of ordinary convolution and hollow convolution with dilation rate of 2

Figures a through c in Figure 4.5 below show how the null convolution works. The upper layer is a convolution kernel of size 3x3 with a step size of 1, and the lower layer is a feature matrix of size 7x7 where the dilation is 2. As with the normal convolutional network algorithm, a sliding window is used for the sampling work. The difference lies in the way the convolutional kernel works, where all points except the center point are sampled, with each expansion coefficient reduced by one row or column accordingly.

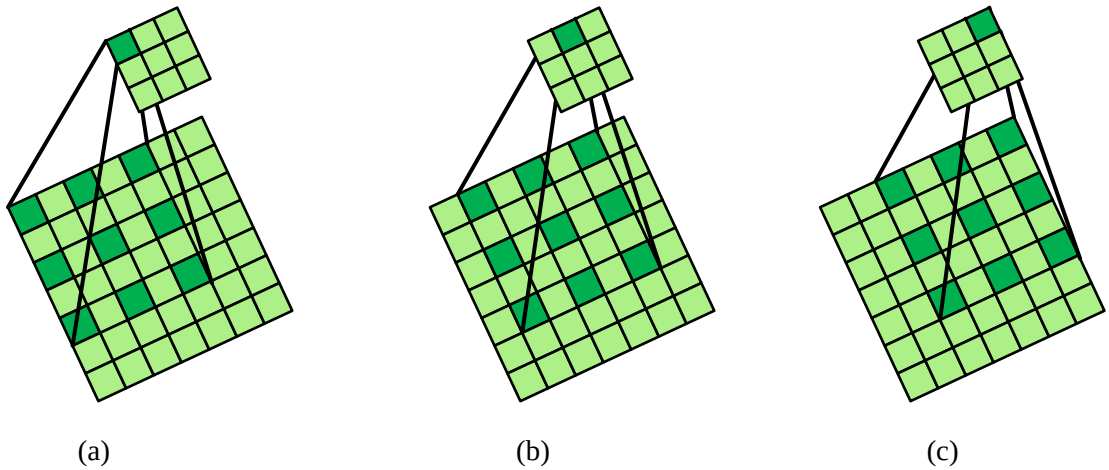

Figure 4.5 Null convolution process

The cavity convolution method is more capable of ensuring the completeness of the spatial semantic information under the same computational conditions as ordinary convolution and without increasing the model complexity. We choose three different sizes of convolution kernels to give it the ability to capture various features, and the computational formula of the cavity convolution algorithm is shown below.

$$d' = W_c \bigoplus_{k=0}^{r} s_{j\pm k} + b \tag{4.13}$$

$$d'' = W_c \bigoplus_{k=0}^{r} s_{i\pm k\sigma} + b \tag{4.14}$$

Where W_c is denoted as the convolution matrix, s_i is the vector representation of the sentence, and s_j is the vector representation of the entity. r is the width of the sliding window, and b is the bias vector. s_i and s_j are convolved by convolution operation to obtain d^' and d^", which are convolved by convolution kernel of one size to obtain $d_{1i} = d' + d''$. Finally the feature vector $d_{ij} = d_{i1} + d_{i2} + d_{i3}(i = 1,2,\dots,n)$ obtained by splicing three different types of feature vectors is obtained after the convolution operation with three different sizes of convolution kernels.

4.3 Experiment and Analysis

This experiment also uses the same evaluation metrics and experimental environment for precision, recall and F1 value as the experiments in Chapter III.

In this experiment, we applied a grid search-based method on the training set to determine the optimal parameters of this model in terms of parameter setting. The grid search method is a method of exhaustive search for parameter values by combining the possible values of each parameter to form a "grid" of parameter values[28]. When traversing the parameter lattice, we often use the cross-validation method to evaluate the model and select the best combination of parameters as the final tuning result. In addition, in this experiment, Adam is chosen as the optimizer to obtain the most suitable learning rate for the parameter of the moment. The final experimental parameters are shown in Table 4.1 below.

Table 4.1 Experimental parameters

parameters	parameter value
learning rate	5e-4
Hidden Layer Dimension	320
Batch_size	8
Dropout	0.5
Epoch	100
Maximum sentence length	128

In this section, the experimental results of the model proposed in this paper are compared with representative models, and the data used for the experiment is the DuIE2.0 dataset after data enhancement in the previous chapter,

which has already proved the effect of data enhancement, and will not be repeated here[29,30]. The results of the comparison experiments for the above models with the models proposed in this paper are shown in Table 4.2 below.

Table 3.2 Comparative test results

model	P/%	R/%	F1/%
ERNIE	66.18	64.51	65.21
BERT-Multi-Head	68.42	68.24	67.98
RSA-PCNN	65.38	63.24	64.26
RTE-BiLSTM	70.19	69.48	69.41

From the results in the above table, it can be seen that compared to the above models, the models proposed in this chapter have the highest F1 values, thus validating the effectiveness of the models in this chapter for the entity relationship extraction task[31]. In comparison to the above two extraction models based on joint modeling, the approach used in this chapter has the following main advantages.

1. The model in this paper adopts ROBERTa-wwm-ext, which is jointly introduced by HIT and KDDI, to obtain the vector representation of sentences. This language model can be more adaptive to the characteristics of Chinese semantics on the one hand, and make more effective use of the long-distance contextual semantic information in the sentences on the other hand.

2. The method of extracting an entity and then passing the extracted entity information into the relationship classification model is a better solution to the problems of entity nesting and relationship overlapping. The performance of the model can also be guaranteed in the case of more entity nesting and relationship overlapping in the dataset.

3. Learning entity-relationship information of triples in sentences through TransE knowledge representation, which can effectively recognize entities and their relationship types even in the face of partial spoken data.

4. Compared with the BERT Multi-Head model, the F1 value of the model proposed in this chapter is improved by 1.85%. This is due to the fact that the used pre-trained model ROBERTa-wwm-ext has better performance on the Chinese domain dataset and the use of Knowledge Representation Learning to enhance the mining of sentence information for relationship classification, which ensures the performance of the model.

5. Conclusion

The machine oral translation system is a concentrated reflection of the level of artificial intelligence, and the oral understanding model, as the core part of computer translation systems, is a research hotspot and difficulty. This article conducts in-depth research on hierarchical evaluation by combining oral signals and deep computer neural networks. The main research results are as follows:

(1) This article conducts experimental research on the robustness of intent recognition based on pre trained oral signal processing on the Snipps dataset and logistics outbound call dataset. Firstly, the shortcomings of pre trained

language models in oral intention recognition scenarios were introduced, and the importance of optimizing pre trained language patterns was elucidated. At the same time, the accuracy of oral signals in establishing grading standards was demonstrated. A pre trained ELMo model based on optimized similar word vectors was reproduced on the English Snipps dataset, and the model and method were applied to outbound logistics data in China.

(2) Given the Transformer baseline model, an additional attention head signal is supervised during the training process, and this dedicated signal is used to obtain better word alignment in real-time during the decoding process, thereby improving the machine translation intervention method based on dictionary guided decoding. The results of numerous experiments indicate that methods based on oral signal processing can achieve steady improvement.

(3) We propose a joint extraction modeling method that includes understanding illustrations based entirely on RoBERTa wwm ext pre trained language human models to alleviate issues such as error propagation. Selecting TransE proprietary technology illustrations in the relationship classification module to study entity relationship statistics in sentences improves the overall performance of the model's extraction. Finally, using a tree shaped multi-layer combination model, a tree shaped two-layer combination model is constructed, which consists of several BP neural networks as the lowest level classifiers and LVQ neural networks as the upper level classification combiner. The DuIE2.0 dataset is trained, and through experimental evaluation of different models, the accuracy, recall, and F1 cost of the models have been improved to a certain extent.

6. References

[1] Huang W, Cheng X, Wang T, et al. Bert-based multi-head selection for joint entity-relation extraction[C]//Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8. Springer International Publishing, 2019: 713-723.

[2] Zhang J, Lin S, Ding L, et al. Multi-scale context aggregation for semantic segmentation of remote sensing images[J]. Remote Sensing, 2020, 12(4): 701.

[3] Chalavadi V, Jeripothula P, Datla R, et al. mSODANet: A network for multi-scale object detection in aerial images using hierarchical dilated convolutions[J]. Pattern Recognition, 2022, 126: 108548.

[4] Sun Y, Wang S, Li Y, et al. Ernie: Enhanced representation through knowledge integration[J]. arXiv preprint arXiv:1904.09223, 2019.

[5] Cui Y, Che W, Liu T, et al. Pre-training with whole word masking for chinese bert. arXiv 2019[J]. arXiv preprint arXiv:1906.08101.

[6] Mandya A, Bollegala D, Coenen F. Contextualised graph attention for improved relation extraction[J]. arXiv preprint arXiv:2004.10624, 2020.

[7] Chen H, Liu X, Yin D, et al. A survey on dialogue systems: Recent advances and new frontiers[J]. Acm Sigkdd Explorations Newsletter, 2017, 19(2): 25-35.

[8] Wei Tingxin, Qu Weiguang, Song Li, et al. A review of complex sentences for abstract semantic representation in Chinese [J]. Journal of Xiamen University (Natural Science Edition), 2018, 57(6).

[9] Wang Y, Wang G, Chen C, et al. Multi-scale dilated convolution of convolutional neural network for image denoising[J]. Multimedia Tools and Applications, 2019, 78: 19945-19960.

[10] Goudjil M, Koudil M, Bedda M, et al. A novel active learning method using SVM for text classification[J]. International Journal of Automation and Computing, 2018, 15: 290-298.

[11] Zhang Chunying, Li Chunhu, LAN Siwu. User intent classification based on multi-granularity feature fusion [J]. Journal of North China University of Science and Technology (Natural Science Edition), 2019.

[12] Lu Ling, Yang Wu, Yang Youjun, et al. Chinese short text classification based on semantic extension and convolutional neural networks [J]. Journal of Computer Applications, 2017, 37(12): 3498-3503.

[13] Majumder N, Poria S, Gelbukh A, et al. Deep learning-based document modeling for personality detection from text[J]. IEEE Intelligent Systems, 2017, 32(2): 74-79.

[14] Wei Pengfei, Zeng Bi, Wang Minghui, et al. A review of joint modeling algorithms for oral comprehension based on Deep learning [J]. Journal of Software, 2021, 33(11): 4192-4216. (in Chinese)

[15] Zhao Kailin, JIN Xiaolong, WANG Yuanzhuo. Review of research on small sample learning [J]. Journal of Software, 2020, 32(2): 349-369. (in Chinese)

[16] Mesnil G, Dauphin Y, Yao K, et al. Using recurrent neural networks for slot filling in spoken language understanding[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2014, 23(3): 530-539.

[17] Feng YH, Yu H, Sun G, et al. Named entity recognition based on BLSTM [J]. Computer Science, 2018, 45(2): 261-268. (in Chinese)

[18] Chinese named entity recognition method based on BERT [J]. Artificial Intelligence and Robotics Research, 2021, 10: 215.

[19] Yao Yu. End-to-end Chinese Speech Recognition System based on bidirectional long and short time memory association timing classification and weighted finite state Converter [J]. Journal of Computer Applications, 2018, 38(9): 2495-2499.

[20] Chen H, Liu X, Yin D, et al. A survey on dialogue systems: Recent advances and new frontiers[J]. Acm Sigkdd Explorations Newsletter, 2017, 19(2): 25-35.

[21] Sak H, Saraclar M, Gungor T. Discriminative reranking of ASR hypotheses with morpholexical and n-best-list features[C]//2011 IEEE Workshop on Automatic Speech Recognition & Understanding. IEEE, 2011: 202-207.

[22] Serdyuk D, Wang Y, Fuegen C, et al. Towards end-to-end spoken language understanding[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5754-5758.

[23] McCann B, Bradbury J, Xiong C, et al. Learned in translation: Contextualized word vectors[J]. Advances in neural information processing systems, 2017, 30.

[24] Zhou J Y M W H, Yu C Z W Z Y, Li L. Towards Making the Most of BERT in Neural Machine Translation[J]. 2020.

[25] Anderson P, Fernando B, Johnson M, et al. Guided open vocabulary image captioning with constrained beam search[J]. arXiv preprint arXiv:1612.00576, 2016.

[26] Li S, He W, Shi Y, et al. Duie: A large-scale Chinese dataset for information extraction[C]//Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8. Springer International Publishing, 2019: 791-800.

[27] Peng Hui, Miao Mengyan, HE Haitao. Task design of Graded Oral English Teaching based on Skehan theoretical framework [J]. Journal of Changsha Railway University: Social Science Edition, 2013 (1): 156-157.

[28] Miwa M, Bansal M. End-to-end relation extraction using lstms on sequences and tree structures[J]. arXiv preprint arXiv:1601.00770, 2016.

[29] Huang W, Cheng X, Wang T, et al. Bert-based multi-head selection for joint entity-relation extraction[C]//Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II 8. Springer International Publishing, 2019: 713-723.

[30] Lan Z, Chen M, Goodman S, et al. Albert: A lite bert for self-supervised learning of language representations[J]. arXiv preprint arXiv:1909.11942, 2019.

[31] Hakkani-Tur D, Tur G, Celikyilmaz A, et al. Multi-domain joint semantic frame parsing using bi-directional rnn-lstm[C]//Interspeech. 2016: 715-719.