END-TO-END CHINESE DIALECT DISCRIMINATION WITH SELF-ATTENTION

Yangjie Dan, Fan Xu*, Mingwen Wang

School of Computer Information Engineering,
Jiangxi Normal University, Nanchang 330022, China

ABSTRACT

Dialect discrimination has an important practical significance for protecting inheritance of dialects. The traditional dialect discrimination methods pay much attention to the underlying acoustic features, and ignore the meaning of the pronunciation itself, resulting in low performance. This paper systematically explores the validity of the pronunciation features of dialect speech composed of phoneme sequence information for dialect discrimination, and designs an end-to-end dialect discrimination model based on the multi-head self-attention mechanism. Specifically, we first adopt the residual convolution neural network and the multi-head self-attention mechanism to effectively extract the phoneme sequence features unique to different dialects to compose the novel phonetic features. Then, we perform dialect discrimination based on the extracted phonetic features using the self-attention mechanism and bi-directional long short-term memory networks. The experimental results on the large-scale benchmark 10-way Chinese dialect corpus released by IFLYTEK[1] show that our model outperforms the state-of-the-art alternatives by large margin.

KEYWORDS

Dialect discrimination, Multi-head attention mechanism, Phonetic sequence, Connectionist temporal classification.

1. INTRODUCTION

With the gradual advancement and promotion of Putonghua, the hometown dialects of many provinces and cities in China have been gradually assimilated by Putonghua. According to statistics from United Nations Educational Scientific and Cultural Organization (UNESCO), a language will disappear every two weeks. However, Chinese dialect, as an excellent intangible cultural heritage of the Chinese nation, should not disappear with the popularization of Mandarin. As a special language variant, Chinese dialect has always been a research hotspot in linguistics. It is urgent to protect dialects.

In 2018, based on the "Dialect Protection Plan", iFLYTEK released the world's first large-scale 10-way precious dialect (Ningxia, Hefei, Sichuan, etc.) phonetic and phoneme corpora covering most parts of my country in order to jointly advance the algorithm research and protection of dialects. Traditional language recognition methods[2] focus on the underlying acoustic features,

* Corresponding author: xufan@jxnu.edu.cn

[1] https://www.iflytek.com/index.html

[2] Language recognition task learns the distinguishing characteristics between different languages through speech sentences and corresponding language tags; dialect recognition is a special case in language recognition. The task of identifying dialect types is to distinguish different language variants in the same language. Compared with distinguishing different languages, distinguishing dialects is more challenging.

David C. Wyld: CSITY, NWCOM, SIGPRO, ASOFT, AIFZ, BDIoT, ITCCMA, CLSB, DTMN, MLNLP - 2021
pp. 305-316, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111425

such as MFCC (mel-frequency cepstral coefficients) and Fbank (log mel-filterbank), without considering the meaning of the pronunciation itself, resulting in poor performance. In fact, when human beings distinguish different types of dialects, they often judge them by the pronunciation characteristics of the dialect itself. More specifically, we investigated the pronunciation dictionary[3] of Chinese dialects to list the phoneme forms of "fang" and "yan" in Minnan dialect, Guangzhou dialect, Hakka dialect and Shanghai dialect as shown in Table 1. It can be seen from Table 1 that the corresponding pronunciation forms (phonemes) of the same Chinese characters in different dialects are completely different. In other words, if we can effectively extract the unique pronunciation features of different dialects, we can use the pronunciation features of dialects to better distinguish different types of dialects.

Table 1. Examples of phonemes in different Chinese dialects.

Dialect type	fang	yan
Minnan dialect	beng1 hng1 hong1	ngian2
Guangzhou dialect	fong1	jin4
Hakka dialect	fong1	ngien2
Shanghai dialect	faon	gni re yi

Based on this observation, this paper systematically explores the effectiveness of the dialect phonetic features composed of phoneme sequence information for language discrimination, and designs an end-to-end dialect discrimination model based on the multi-head self-attention mechanism. The model first adopts residual CNN (convolutional neural networks)[11] and multi-head attention mechanism to effectively extract the unique phoneme sequence information of different dialects to generate voice pronunciation features, and then uses attention mechanism and bidirectional long short-term memory (BiLSTM)[18] for dialect discrimination. We conducted experiments on the large-scale benchmark 10-way dialect corpus released by iFlytek. The experimental results show that the multi-headed self-attention mechanism [30] can effectively extract the pronunciation characteristics of phoneme sequences unique to different dialects, which greatly improves the discrimination performance of dialects.

The follow-up content of this paper is organized as follows: Section 2 presents the related work of language discrimination in recent years; Section 3 illustrates our model in detail; Section 4 introduces the data set, experimental settings and detailed analysis of experimental results; Section 5 concludes the paper.

2. RELATED WORK

This section mainly introduces representative language discrimination models from two perspectives: traditional acoustic features based and speech pronunciation features based models.

2.1. Methods based on Traditional Acoustic Features

Traditional language discrimination methods adopt underlying acoustic features to build acoustic models in order to obtain fixed encoding vectors of speech sentences. At present, the commonly used artificially extracted underlying acoustic features include: Fbank (log mel-filterbank), MFCC (mel-frequency cepstral coefficients), PLP (perceptual linear prediction), SDC (delta coefficients) [1,2], etc. Since the underlying acoustic features are extracted in units of frames, the number of frames corresponding to speech sentences with different durations is also different.

[3] http://cn.voicedic.com/

Therefore, how to convert a variable-length speech sentence into a fixed vector representation is a vital step. The typical methods are GMM (Gaussian mixture model) super vector [3] and GMM i-vector [4]. The i-vector feature contains relevant information about the speaker and language. This feature is usually used as a speech sentence representation to train a language classifier. Commonly used classifiers include multi-class logistic regression and support vector machines. But the main disadvantage of the i-vector method is its poor discrimination effect on short speech sentences [5].

Recently, as deep learning technology has achieved great success in speech recognition tasks[6], some researchers have begun to explore language discrimination technology based on deep learning. In the early days, many studies [7,8,9] used deep learning technology to extract the bottleneck features of speech sentences, and achieved better language discrimination performance. Currently, some researchers recognize the powerful representation capabilities of deep learning and directly use various types of neural networks to build end-to-end language discrimination models. It was first used by Lopez-Moreno et al.[5] to successfully use deep neural networks for language discrimination. The network directly takes the underlying acoustic features of the speech sentence, and then scores each frame on different languages. The score of the speech sentence is the average of all frames within the sentence. After that, there are many language discrimination models with different structures, such as deep neural networks (DNNs) based on attention mechanism [10], convolutional neural networks (CNNs) [11,12,13,14], delayed neural networks [15,16] and recurrent neural networks (RNNs). Because the RNN network has a strong ability to extract context-related (global) features, it can learn better feature representations for the temporal feature characteristics of speech, which improves the performance of the language discrimination. In practical, there are several different variants of recurrent neural networks, including gated recurrent unit recurrent neural network (GRU) [17], long and short-term memory recurrent neural network [18,19,20,21,22], Bidirectional long and short-term memory recurrent neural network [23,24].

2.2. Method Based on Pronunciation Characteristics

Traditional language discrimination models based on underlying acoustic features ignore many important speech pronunciation information. Therefore, Tang et al. [25] adopted the pronunciation characteristics of speech to improve the effect of language discrimination. The specific method was to use a speech phoneme recognition model to extract the frame-level speech pronunciation characteristics, and then feed the speech pronunciation characteristics into the language discrimination model. The speech phoneme recognition model of this method uses a cross-entropy loss function. Recently, studies have shown that the end-to-end acoustic model based on CTC [26] (connectionist temporal classification) has obtained better performance [27, 28, 29].

Based on these observation, this paper proposes an end-to-end dialect discrimination model based on multi-head self-attention mechanism. We adopt the CTC loss function to train the acoustic model of speech phoneme recognition, and integrate a multi-headed self-attention layer, which can give the acoustic model the ability to extract unique pronunciation characteristics of different dialects. The multi-head self-attention mechanism is based on the transformer [30] model proposed by Google in 2017. This model performs very well on machine translation, and it also has a good effect on speech recognition [31] tasks.

3. SELF-ATTENTION DRIVEN DIALECT DISCRIMINATION

Figure 1 illustrates the proposed dialect discrimination model. The model is mainly composed of two parts. The top side of the figure is the speech phoneme recognition model, and the main function of this part is to extract the pronunciation characteristics of the dialect. The bottom side of the figure is the dialect discrimination model, which mainly uses the pronunciation characteristics of dialects to improve the accuracy of dialect discrimination. In the speech phoneme recognition model, we extract more abstract local features of the speech through a residual CNN, and then feed them into a multi-headed self-attention layer, which can pay attention to the relationship between each frame of speech and other frames, and then map to the appropriate dimension through a fully connected layer, and finally calculate the difference between the predicted phoneme sequence and the real phoneme sequence through the CTC loss function. When identifying dialect types, we designed two models. One is to input the recognized dialect pronunciation features into Self-Attention Pooling (SAP) [34]. This attention mechanism can encode the variable-length dialect pronunciation features into a fixed vector representation, which is then input to the fully connected layer. In fact, we can also do an average pooling or maximum pooling of pronunciation features, but the attention mechanism is essentially a weighted average operation on the pronunciation features, and the weighted average can include these two situations according to the different proportions of the allocation. Then, we feed the recognized pronunciation features of dialects into the BiLSTM network. We use the output of the last moment of BiLSTM as the fixed-length vector representation of the speech sentence, then it is mapped to 10 dialects through two fully connected layers. The first fully connected layer maps the input features to each implicit semantic node, and the second fully connected layer represents the display expression of the classification. Finally, the probability of the speech sentence belonging to each dialect is obtained through softmax. The function of each sub-module is introduced below.

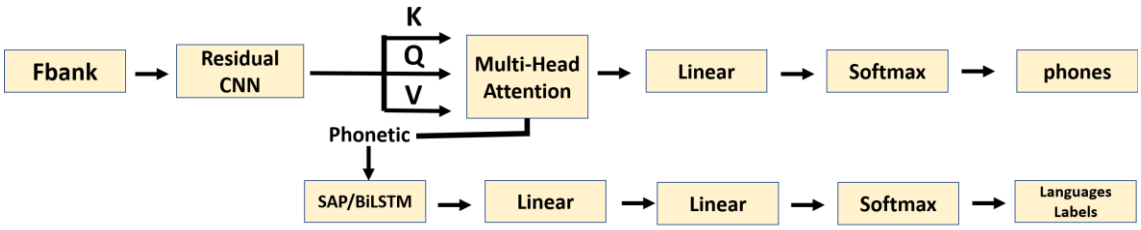

Figure 1. Dialect discrimination model based on self-attention mechanism

3.1. Residual CNN

Residual network [32] was first applied to image classification, and Li et al. [33] adopted residual CNN to extract speech features and performed language discrimination. In fact, CNN can better extract features on voice frequency, and residual CNN can use a deeper network to extract more abstract voice features. Due to the existence of the residual mechanism, even if the number of network layers increases, it will not cause network degradation. In order to obtain a more abstract representation of the speech sentence, we design the residual CNN network structure based on resnet18.

3.2. Self-Attention Mechanism

The self-attention mechanism is a coding sequence scheme proposed by the Google team Vaswani et al. [30] in 2017. It can be considered that it is a sequence coding layer like general

CNN and RNN. The self-attention mechanism is a special attention mechanism, and it only needs a separate sequence to calculate the code of this sequence. The self-attention mechanism uses standard dot product attention, and its calculated attention weight is shown in formula (1):

$$Attention(\boldsymbol{Q},\boldsymbol{K},\boldsymbol{V}) = softmax(\frac{\boldsymbol{Q}\boldsymbol{K}^{\boldsymbol{T}}}{\sqrt{d_k}})\boldsymbol{V} \quad (1)$$

where the dimensions of the query vector *Q* and the key vector *K* are both d_k, and the length of the value vector *V* is d_v.

The multi-head self-attention mechanism is adopted to calculate a single attention multiple times, but the query vector, key vector, and value vector are different each time. Specifically, the multi-head self-attention mechanism layer first generates *h* different query vectors *Q*, key vectors *K*, and value vectors *V*, where the dimensions of the query vector and the key vector are d_k, and the dimension of the value vector is d_v. For each set of query vectors, key vectors and value vectors, a vector with dimension d_v can be generated by formula (1), and then the generated h vectors can be spliced together. The above process can be described by formula (2) and formula (3):

$$\text{MultiHead}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \text{Concat}(head_1,\dots,head_h)\boldsymbol{W}^{\boldsymbol{O}} \quad (2)$$

$$head_i = Attention(\boldsymbol{Q}\boldsymbol{W}_i^{\boldsymbol{Q}},\boldsymbol{K}\boldsymbol{W}_i^{\boldsymbol{K}},\boldsymbol{V}\boldsymbol{W}_i^{\boldsymbol{V}}) \quad (3)$$

where $\boldsymbol{W}_i^{\boldsymbol{Q}} \epsilon R^{d_{model}\times d_k}$, $\boldsymbol{W}_i^{\boldsymbol{K}} \epsilon R^{d_{model}\times d_k}$, $\boldsymbol{W}_i^{\boldsymbol{V}} \epsilon R^{d_{model}\times d_v}$, $\boldsymbol{W}_i^{\boldsymbol{O}} \epsilon R^{hd_v\times d_{model}}$, *h* is the number of heads, d_{model} is the model dimension.

3.3. Connectionist Temporal Classification (CTC)

The CTC loss function can select the sequence with the greatest probability from the given input sequence [26]. We use *x* to represent the input sequence and *z* to represent the corresponding phoneme sequence. Each training sample can be represented by a tuple (x, z). Our goal of maximizing the likelihood function is to minimize the value of formula (4).

$$\mathcal{L}_{ctc} = -ln\prod_{(\boldsymbol{x},\boldsymbol{z})\epsilon S} p(\boldsymbol{z}|\boldsymbol{x}) = -\sum_{(\boldsymbol{x},\boldsymbol{z})\epsilon S} lnp(\boldsymbol{z}|\boldsymbol{x}) \quad (4)$$

where $(x, z) \in s$ represents a training sample.

3.4. Bidirectional Long Short-Term Memory Network

LSTM is a special recurrent neural network, which has the ability to learn long-term dependence, and is suitable for processing and predicting important events with relatively long intervals and delays in time series. The relevant parameter calculation formula of the LSTM model is as follows:

$$\boldsymbol{i}_i = sigmoid(\boldsymbol{W}_i[\boldsymbol{x}_i,\boldsymbol{h}_{i-1}] + \boldsymbol{b}_i) \quad (5)$$

$$\widetilde{\boldsymbol{c}}_i = \tanh(\boldsymbol{W}_c[\boldsymbol{x}_i,\boldsymbol{h}_{i-1}] + \boldsymbol{b}_c) \quad (6)$$

$$\boldsymbol{f}_i = sigmoid(\boldsymbol{W}_f[\boldsymbol{x}_i,\boldsymbol{h}_{i-1}] + \boldsymbol{b}_f) \quad (7)$$

$$\boldsymbol{o}_i = sigmoid(\boldsymbol{W}_o[\boldsymbol{x}_i,\boldsymbol{h}_{i-1}] + \boldsymbol{b}_o) \quad (8)$$

$$\boldsymbol{c}_i = \boldsymbol{i}_i * \widetilde{\boldsymbol{c}}_i + \boldsymbol{f}_i * \boldsymbol{c}_{i-1} \quad (9)$$

$$\boldsymbol{h}_i = \boldsymbol{o}_i * tanh(\boldsymbol{c}_i) \quad (10)$$

From the parameter calculation formula of the LSTM model, it is known that the LSTM unit has three "gate" structures that determine the state of the cell. They are input gate, forget gate and output gate. The value of the sigmoid function is between 0 and 1. As the gate structure and the input data are multiplied, the amount of information of the input data can be determined. The input variables that determine the states of these three doors are the same, but the parameters corresponding to the doors of different functions are different. It can be seen from the formula that the states of the three gate structures at a certain moment are all related to the current input x_i and the output value h_{i-1} at the previous moment. The parameters that determine the state of the forget gate are W_f and b_f. The state determines the proportion of the previous state information that is forgotten at the current moment. The parameters that determine the state of the output gate are W_o and b_o, and the state determines the ratio of the current time and the previous state information. The parameters that determine the state of the input gate are$\boldsymbol{W}_i$ and $\boldsymbol{b}_i$, and this state determines how much information input at the current moment is retained. The BiLSTM takes the order of the sequence into account, that is, there are two layers of one-way LSTM, one layer extracts the global features of the order, and the other layer extracts the global features of the reverse order.

4. EXPERIMENT AND RESULT ANALYSIS

This section mainly describes the iFLYTEK dialect data set, baseline models, parameter settings, and analysis of experimental results.

4.1. Data set

The dialect speech data set used in this experiment was released by iFLYTEK. It contains 10 different dialect speech and phoneme corpora[4]. The statistics of the corpus is shown in Table 2. Each dialect contains an average of 6 hours of reading style speech data, covering 35 people.

Table 2. Statistics of the iFLYTEK dialect data set.

Data set	**Training set**			**Development set**		
Dialect	Speaker	Sentence per person	Total sentences	Speaker	Sentence per person	Total sentences
Ningxia dialect	30	200	6000	5	100	500
Hefei dialect	30	200	6000	5	100	500
Sichuan dialect	30	200	6000	5	100	500
Shanxi dialect	30	200	6000	5	100	500
Changsha dialect	30	200	6000	5	100	500
Hebei dialect	30	200	6000	5	100	500
Nanchang dialect	30	200	6000	5	100	500
Shanghai dialect	30	200	6000	5	100	500
Hakka dialect	30	200	6000	5	100	500
Minnan dialect	30	200	6000	5	100	500

[4] http://challenge.xfyun.cn/2018/aicompetition/tech

The data was collected by various types of smart phones, and the recording environment includes a quiet environment and a noisy environment. The data was stored in a PCM format with a sampling rate of 16000 Hz and 16-bit quantization. The data set contains training set and development set. There are 60,000 sentences in the training set, 6000 sentences in each dialect, including 30 speakers, 15 males and 15 females, and 200 voices per speaker; the development set has 5000 voices, each dialect has 500 voices, and each dialect contains 5 speakers, including 2 females and 3 males. Each phonetic sentence also has its corresponding phoneme label, such as "l iou4 sh iii2 _e er4 _v van2 s ii4 f en1". We adopt 60,000 speech sentences as our training set, and take 5000 speech sentences as our testing set.

4.2. Baseline Models

We adopt three benchmark models for dialect discrimination. The first one is a dialect discrimination model based on i-vector features, the second is the LSTM-based dialect discrimination model officially provided by iFLYTEK[5], and the third is the single model adopted by the first winner of the first dialect discrimination competition in 2018[6].

Baseline model 1: This model proposed a dialect discrimination model based on i-vector features. More specifically, 60-dimensional MFCC features were extracted, which include first-order and second-order difference coefficients. The general background model used to extract i-vector features includes 2048 Gaussian functions, and finally 400-dimensional i-vector features are extracted from each sentence. The baseline model also uses a support vector machine as a classifier.

Baseline model 2: This model is officially provided by iFLYTEK. It uses a one-way LSTM and two fully connected layers. The hidden unit of the LSTM has a dimension of 128, and the input dimension of the first layer of fully connected layer is 128. The output dimension is 30. The input dimension of the second fully connected layer is 30, and the final output dimension is 10.

Baseline model 3: This model is a single model presented by the first winner of the 2018 first dialect discrimination competition of iFLYTEK. The model is divided into speech phoneme recognition model and dialect discrimination model. The speech phoneme recognition model uses residual CNN and BiLSTM. The dialect discrimination model is to fix the parameters of the trained residual CNN model to remain unchanged, and then add a layer of trainable BiLSTM, and finally integrate the output states of the BiLSTM at all times, that is, use the output state at all times. In short, both the speech phoneme recognition model and the dialect discrimination model have a residual CNN module and their BiLSTM module. For fair comparison, the residual CNN network structure used in this paper is the same as that of this baseline model.

4.3. Parameter Settings

We first seperate the original speech sentence into different frames. The window size of the framing is 25ms, and the frame shift is 10ms. Then we use Kaldi[7] toolkit to extract 80-dimensional Fbank features. The relevant parameter settings of residual CNN are shown in Table 3.

[5] http://bbs.xfyun.cn/forum.php?mod=viewthread&tid=39141

[6] http://1024.iflytek.com/h5/?from=singlemessage；The reason why we adopted the first author single system in the competition is that the final system is a composite model, but the composite model is not disclosed. Moreover, 90.50% of the officially announced recognition performance was obtained on the undisclosed final competition test set.

[7] http://www.kaldi-asr.org/

Table 3. Parameter settings in residual CNN network.

layer	output size	down sample	channels	blocks
conv1	$40 \times L_{in}/2$	True	64	-
maxpool	$20 \times L_{in}/4$	True	64	-
res1	$10 \times L_{in}/4$	False	64	2
res2	$5 \times L_{in}/4$	False	128	2
res3	$3 \times L_{in}/4$	False	256	1
res4	$2 \times L_{in}/4$	False	512	1
avgpool	$1 \times L_{in}/4$	False	512	-
reshape	$512 \times L_{in}/4$	-	-	-

The parameter settings used by the multi-head self-attention layer are shown in Table 4.

Table 4. Parameter settings in self-attention layer network.

Model	N	d_{model}	h	d_k	d_v
Multi-head	1	512	8	64	64

where *N* in Table 4 represents the number of layers, and other parameters correspond to formula (2) and formula (3). The hidden state of the BiLSTM is 256 dimensions, and the two-way total has 512 dimensions. BiLSTM is followed by two fully connected layers. The first fully connected layer maps 512 dimensions to 256 dimensions, and the second layer maps 256 dimensions to 10 dimensions. The model in this paper uses the Adam optimization algorithm based on the mini-batch gradient descent algorithm. The optimization algorithm can change the learning rate during the training process and control the step length along the gradient descent through the attenuated learning rate. For the speech phoneme discrimination model, we set a learning rate of 0.0005, and for the language discrimination model, we set a learning rate of 0.001. We train 10,000 frames of speech at the same time each time. This article uses the pytorch framework to implement all network models.

For the evaluation indicators, we use three evaluation indicators to describe the discrimination performance of the system, namely: Accuracy (ACC), Average Decision Cost Function (Cavg) and Equal Error Rate (EER). where the accuracy rate is the evaluation index defined by the IFLYTEK Dialect Competition (the ratio of the number of correct speech sentences to the total number of sentences). Average detection cost and equal error rate are the evaluation indicators used in the standard evaluation of NIST LRE [35].

4.4. Result

Table 5 shows the comparison of dialect discrimination performance under various models. It can be seen from the experimental results that the effect of LSTM is slightly worse than that of i-vector, because the discrimination effect of i-vector on short speech (for example, within 3s) is relatively poor [5], but the discrimination effect on relatively long speech is relatively good, and LSTM may not be suitable for processing relatively long speech in the test set [36]. Since BiLSTM can extract context-related features, we use the output vector of the last moment state as a fixed vector representation of a speech sentence. It can be seen in Table 5 that the two models proposed in this paper are better than baseline model 3. In addition, Cavg and EER have also been greatly improved (the smaller the value, the better the performance). Compared with baseline model 3, our model has an extra layer of self-attention. We believe that the self-attention layer can better extract the local part of the speech pronunciation.

Table 5. Performance Comparison of Different Dialect Discrimination Models.

Model	Acc（%）	C_{avg}*100	EER（%）
i-vector (baseline1)	74.30	9.99	10.04
LSTM (baseline2)	74.28	14.03	14.76
iFLYTEK 2018 Dialect Competition First List Model (baseline3)	86.62	7.43	12.98
Our model (the right side in Figure 1 uses the SAP sub-module)	87.34	6.89	5.46
Our model (the BiLSTM sub-module is used on the right in Figure 1)	89.22	5.86	4.8

Since we adopt the characteristics of dialect pronunciation as the input of the dialect discrimination model, we further compare the phoneme recognition performance of these models. We use a greedy algorithm to decode speech into phoneme sequences. The experimental results are shown in Table 6. WER in the table represents the phoneme error rate. The lower the WER, the better the effect of the speech phoneme recognition model. The WER obtained by our model is higher than the first place system in the iFLYTEK competition. Regarding this phenomenon, we believe that the multi-head self-attention mechanism can better extract the unique pronunciation characteristics of different dialects, and the BiLSTM of the first place system in the iFLYTEK competition is more suitable to extract the pronunciation characteristics commonly shared by different dialects. Therefore, in contrast, the pronunciation features extracted by the multi-head self-attention mechanism are more discriminative in the discrimination of dialect types.

Table 6. Speech phoneme recognition performance comparison.

Model	WER（%）
Residual CNN [32]	46.57
Our model (after adding multi-head)	43.08
The phoneme recognition model of iFLYTEK 2018 Dialect Competition No. 1	41.06

In order to verify the results, we designed a unique dialect phoneme recognition discrimination experiment as shown in Table 7. First, we use the 60,000 phoneme sentences of the training set to train an SVM classifier, which inputs sentence phoneme sequences and outputs dialect types. When testing, we use ASR1 (residual CNN+Multi-Head Attention + CTC) and ASR2 (residual CNN+BiLSTM +CTC), where ASR stands for Automatic Speech Recognition, and the identified phoneme sequence is tested. We first count the unary language models of 10 different dialects in the training set. We believe that the higher the frequency of phonemes in different dialects, the more representative the dialect. We extracted phonemes with word frequency greater than 1%, 0.9%, and 0.8% as features, and we regarded all other phonemes as unregistered words. The word frequency is greater than 1%, 0.9%, and 0.8% have 33, 40 and 47 phonemes, respectively. There are 5000 sentences in the test set.

Table 7. Distinguishing Experiments on Phoneme Recognition of Unique Dialects

	1%(33)	0.9%(40)	0.8%(47)
ASR1	907	936	942
ASR2	904	925	973

We found that at 0.9% of the time, 40 phonemes were selected as features, and the recognition effect on ASR1 was better than ASR2, indicating that the multi-headed self-attention mechanism recognized more dialect-specific phonemes.

CASE STUDY: Figure 2 and 3 show two instances (example 1 and 2 represented as phoneme respectively; the number 1, 2, 3 and 4 indicate four lexical tones of Chinese).

Example 1: m ei2 _i ia1 b u2 sh iii4 _u uo3 z ai4 n a4 n i3 p ei2 _u uo3 l iao2 m a1
Example 2: g uo2 j ia1 b o2 _u u4 g uan3 l ao3 d a4 l ao3 d a4

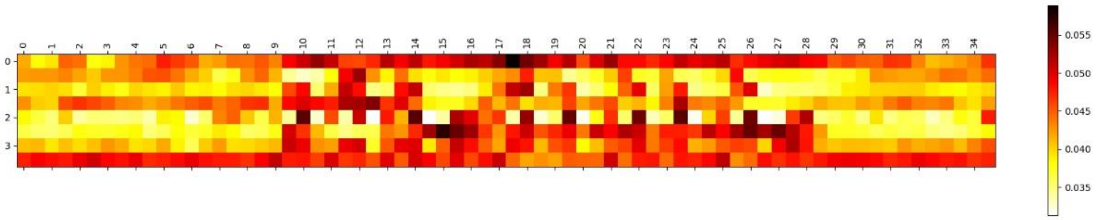

Figure 2. Attention visualization for an instance from shanghai dialect.

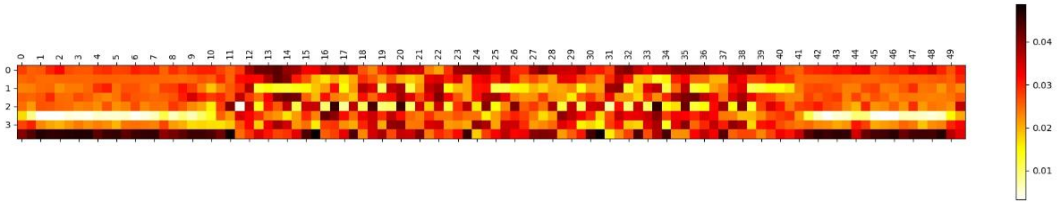

Figure 3. Attention visualization for an instance from sichuan dialect.

As shown, we can observe the attention is useful to conduct Chinese dialects discrimination. For Figure 2, this audio file is an example in shanghai dialect, and the number of frames is 280. After handling by our model, the number of frames is extracted into 70. The x-coordinate 0-34 represents the extracted 70 frames with interval 2, and the y-coordinate represents the 8 heads of the attention mechanism with interval 2. It can be seen that each head has a certain degree of access to audio information, and the discriminative phoneme of the frame number range from 20 to 56 have a great impact. Similarly, for Figure 3, this audio file is an example in sichuan dialect, the number of frames is 400. After handling by our model, the number of frames is extracted into 100 frames. The x-coordinate 0-49 represents 40 frames after extraction with interval 2, and the y-coordinate represents 8 heads of the attention mechanism with interval 2. It can be seen that the last one heads get the most information (with the deepest colour), and the features with frame number ranging from 26-76 have a great influence.

Although our model has achieved better performance in language discrimination tasks, the word error rate of our model in dialect speech recognition is still quite low. In the future, we will improve the performance of dialect speech recognition model through integrating more dialect corpus.

5. CONCLUSIONS

This paper designs an end-to-end dialect discrimination model based on multi-headed self-attention mechanism, which considers the influence of dialect pronunciation characteristics (phoneme sequence). In terms of dialect pronunciation features, we compared the pronunciation

features extracted from different model structures. The experimental results on the benchmark speech corpus of 10 major dialects released by iFLYTEK demonstrate the effectiveness of the multi-headed self-attention mechanism, the performance of dialect discrimination has been greatly improved. We will further study how to better extract the unique pronunciation features of different dialects and design a composite model to further improve the performance of dialect discrimination. In the future, we will also expand our dialect corpus and focus on improving the performance of dialect speech recognition.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for their insightful comments on this paper. This research was supported by the National Natural Science Foundation of China (NSFC) under Grant 61772246 and 61876074, Natural Science Foundation of Jiangxi Province under Grant 20192ACBL21030.

REFERENCES

[1] Kinnunen T, Li H. An overview of text-independent speaker recognition: From features to supervectors[J]. Speech communication, 2010, 52(1): 12-40.
[2] Li H, Ma B, Lee K A. Spoken language recognition: from fundamentals to practice[J]. Proceedings of the IEEE, 2013, 101(5): 1136-1159.
[3] Campbell W M, Sturim D E, Reynolds D A. Support vector machines using GMM supervectors for speaker verification[J]. IEEE signal processing letters, 2006, 13(5): 308-311.
[4] Dehak N, Kenny P J, Dehak R, et al. Front-end factor analysis for speaker verification[J]. IEEE Transactions on Audio, Speech, and Language Processing, 2010, 19(4): 788-798.
[5] Lopez-Moreno I, Gonzalez-Dominguez J, Plchot O, et al. Automatic language identification using deep neural networks[C]//2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2014: 5337-5341.
[6] Dahl G E, Yu D, Deng L, et al. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition[J]. IEEE Transactions on audio, speech, and language processing, 2011, 20(1): 30-42.
[7] Richardson F, Reynolds D, Dehak N. A unified deep neural network for speaker and language recognition[J]. arXiv preprint arXiv:1504.00923, 2015.
[8] Ferrer L, Lei Y, McLaren M, et al. Study of senone-based deep neural network approaches for spoken language recognition[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2015, 24(1): 105-116.
[9] McLaren M, Ferrer L, Lawson A. Exploring the role of phonetic bottleneck features for speaker and language recognition[C]//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5575-5579.
[10] Mounika K V, Achanta S, Lakshmi H R, et al. An Investigation of Deep Neural Network Architectures for Language Recognition in Indian Languages[C]//INTERSPEECH. 2016: 2930-2933.
[11] Lozano-Diez A, Zazo-Candil R, Gonzalez-Dominguez J, et al. An end-to-end approach to language identification in short utterances using convolutional neural networks[C]//Sixteenth Annual Conference of the International Speech Communication Association. 2015.
[12] Ma J, Song Y, McLoughlin I V, et al. LID-senone extraction via deep neural networks for end-to-end language identification[J]. 2016.
[13] Jin M, Song Y, McLoughlin I V, et al. End-to-end language identification using high-order utterance representation with bilinear pooling[J]. 2017.
[14] Cai W, Cai Z, Zhang X, et al. A novel learnable dictionary encoding layer for end-to-end language identification[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5189-5193.
[15] Garcia-Romero D, McCree A. Stacked Long-Term TDNN for Spoken Language Recognition[C]//INTERSPEECH. 2016: 3226-3230.
[16] Tkachenko M, Yamshinin A, Lyubimov N, et al. Language identification using time delay neural network d-vector on short utterances[C]//International Conference on Speech and Computer. Springer, Cham, 2016: 443-449.

[17] Pešán J, Burget L, Černocký J. Sequence summarizing neural networks for spoken language recognition[J]. Interspeech 2016, 2016: 3285-3288.
[18] Gonzalez-Dominguez J, Lopez-Moreno I, Sak H, et al. Automatic language identification using long short-term memory recurrent neural networks[C]//Fifteenth Annual Conference of the International Speech Communication Association. 2014.
[19] Geng W, Wang W, Zhao Y, et al. End-to-end language identification using attention-based recurrent neural networks[J]. 2016.
[20] Gelly G, Gauvain J L. Spoken Language Identification Using LSTM-Based Angular Proximity[C]//INTERSPEECH. 2017: 2566-2570.
[21] Masumura R, Asami T, Masataki H, et al. Parallel phonetically aware DNNs and LSTM-RNNs for frame-by-frame discriminative modeling of spoken language identification[C]//2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2017: 5260-5264.
[22] Tang Z, Wang D, Chen Y, et al. Phonetic temporal neural model for language identification[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 26(1): 134-144.
[23] Gelly G, Gauvain J L, Le V B, et al. A Divide-and-Conquer Approach for Language Identification Based on Recurrent Neural Networks[C]//INTERSPEECH. 2016: 3231-3235.
[24] Fernando S, Sethu V, Ambikairajah E, et al. Bidirectional Modelling for Short Duration Language Identification[C]//INTERSPEECH. 2017: 2809-2813.
[25] Tang Z, Wang D, Chen Y, et al. Phonetic temporal neural model for language identification[J]. IEEE/ACM Transactions on Audio, Speech, and Language Processing, 2017, 26(1): 134-144.
[26] Graves A, Mohamed A, Hinton G. Speech recognition with deep recurrent neural networks[C]//2013 IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 6645-6649.
[27] Miao Y, Gowayyed M, Metze F. EESEN: End-to-end speech recognition using deep RNN models and WFST-based decoding[C]//2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, 2015: 167-174.
[28] Kim S, Hori T, Watanabe S. Joint CTC-attention based end-to-end speech recognition using multi-task learning[C]//2017 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2017: 4835-4839.
[29] Yi J, Tao J, Bai Y. Language-invariant Bottleneck Features from Adversarial End-to-end Acoustic Models for Low Resource Speech Recognition[C]//ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 6071-6075.
[30] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
[31] Zhou S, Dong L, Xu S, et al. Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese[J]. arXiv preprint arXiv:1804.10752, 2018.
[32] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[33] Cai W, Cai Z, Zhang X, et al. A novel learnable dictionary encoding layer for end-to-end language identification[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 5189-5193.
[34] Cai W, Chen J, Li M. Exploring the encoding layer and loss function in end-to-end speaker and language recognition system[J]. arXiv preprint arXiv:1804.05160, 2018.
[35] “The 2015 NIST language recognition evaluation plan (LRE15),” NIST, 2015, ver. 22-3.
[36] Cai W, Cai Z, Liu W, et al. Insights into end-to-end learning scheme for language identification[J]. arXiv preprint arXiv:1804.00381, 2018.