Frontiers in Computer Vision and Pattern Recognition (2023)
Clausius Scientific Press, Canada
DOI: 10.23977/fcvpr.2023.010101
ISSN 2522-6665 Vol. 1 Num. 1

Applying Transfer Learning for Syllable-Based Speech Recognition in Tibetan Language

Senyan Li, Guanyu Li*, Sirui Li

Northwest Minzu University, Lanzhou, Gansu, 730000, China
**Corresponding author*

Keywords: Speech recognition, Tibetan language, low-resource, transfer learning, Amdo dialect

Abstract: This article mainly explores Tibetan speech recognition and reviews its development history. In recent years, end-to-end methods have been applied to Tibetan speech recognition. However, due to the lack of training data, the performance of the end-to-end method is not ideal. Therefore, this article introduces the transfer learning method, which uses Mandarin as a same-language family language to train a pre-trained model that initializes the Tibetan speech recognition model. On the xbmu-amdo31 Tibetan public dataset, our method achieved an 11.8% relative reduction in phoneme error rate compared to the baseline system. This method not only enhances the performance of speech recognition in low-resource languages but also has the potential to be extended to other same-language family languages. Overall, this article highlights the importance of transfer learning in speech recognition and its potential impact on improving speech recognition systems in low-resource languages.

1. Introduction

Speech is the simplest and most commonly used way of human communication. Enabling machines to understand human speech is of great importance for human-machine interaction. Automatic speech recognition (ASR) is a technology that aims to make machines understand human speech and convert it into corresponding text sequences. With the development of deep learning technology, speech recognition has maintained a high-speed development trend over the past decade. There are two main frameworks for speech recognition: hybrid architecture and end-to-end architecture. In recent years, with the continuous progress and performance improvement of the end-to-end architecture, it has gradually become the mainstream method. After decades of development, speech recognition has reached a level comparable to that of humans for resource-rich languages such as Mandarin, English, and German, including speech recognition rate and speech interaction applications. However, research on speech recognition for Tibetan language started slowly around 2005.

So far, the effect of Tibetan speech recognition is still not ideal, mainly because it is difficult to obtain Tibetan audio and perform text annotation. Although some speech databases have been developed, most of them are small-scale data corpora. In the early research on Tibetan speech recognition, dynamic time warping [1] (DTW) algorithm was mainly used for isolated word

recognition. However, DTW algorithm has many limitations. Although its implementation is relatively simple, it has high requirements for endpoint detection technology and has high dependence. Therefore, researchers gradually began to use Hidden Markov Model [2] (HMM)-based methods for Tibetan speech recognition. Since 2009, researchers have started to implement Tibetan isolated word speech recognition based on HMM [3]. As research deepened, some improvement measures were proposed, such as feature enhancement based on resonant peak parameter extraction, statistics and analysis [4,5]. These improvement measures can be implemented at the feature level or the acoustic model level, and can further improve the effect of Tibetan speech recognition. Since 2016, researchers have started to apply neural networks to Tibetan speech recognition and combine them with HMM to further improve recognition performance [6]. Although there have been some minor improvements in Tibetan speech recognition, there is currently no better method in practice.

Since the advent of end-to-end speech recognition in 2014, it has become the mainstream method of speech recognition. The input of this method is acoustic features, which directly output labels. Currently, there are three main structures: Connectionist Temporal Classification [7] (CTC), Recurrent Neural Network-Transducer [8] (RNN-T), and Attention-based Encoder-Decoder [9-12] (AED). For Tibetan speech recognition, this is both an opportunity and a challenge. The end-to-end method can make up for the lack of a standard pronunciation dictionary for Tibetan, but it requires support from a large amount of data, which is also a problem faced by Tibetan speech recognition. In order to solve this problem, this paper draws inspiration from the idea of transfer learning and attempts to use more resource-rich languages to assist in training Tibetan speech recognition models, thereby obtaining better recognition performance.

Currently, the Conformer [13] model has been used in speech recognition. This model not only uses attention mechanism to focus on global features but also introduces convolutional modules to focus on local features, so it performs well in speech recognition. Considering the excellent performance of the Conformer model, this paper uses it as the encoder of the attention-based encoder-decoder model for Tibetan speech recognition tasks, while its decoder is the same as the Transformer [14] model decoder.

This paper is organized as follows. Section 2 will provide an overview of related work, Section 3 will provide a detailed description of the method proposed in this article, followed by the presentation of experimental results in Section 4, and finally, a summary of this work will be provided in Section 5.

2. Related Work

This paper uses an attention-based encoder-decoder approach, which effectively solves the sequence-to-sequence problem. This method does not require pre-segmentation and alignment of data. Through the attention mechanism, it can implicitly learn the soft alignment between the input sequence and the output sequence, thus addressing a major issue in speech recognition. The encoded results generated by this method are no longer limited to a single fixed-length vector. Therefore, the model can handle speech inputs of various lengths and still produce good results.

However, using the attention mechanism alone is not sufficient to achieve ideal performance because in speech recognition, speech and output labels are usually monotonically aligned. Therefore, in order to better align speech features and label sequences, current practice often uses shared encoder attention mechanisms and CTC models for joint optimization in a multitask learning framework. The CTC model can enforce monotonous alignment, thereby improving the convergence of attention-based models and alleviating alignment problems. Currently, this training strategy has become a standard training approach for attention-based speech recognition models.

2.1. Conformer Model

The architecture of the Conformer model is derived from the evolution of the Transformer model encoder architecture. Compared with the Transformer model, the Conformer model not only relies on attention mechanisms and positional encoding but also introduces convolutional modules. The main structure of the Conformer model is composed of multiple stacked blocks. The input to the encoder is speech features, and the input sequence symbol is mapped to $X = (x_1, ..., x_n)$. After the input X is processed by the encoder, a series of output mapping symbols $H = (h_1, ..., h_n)$ are obtained.

2.1.1. Multi-head Attention Mechanism

The Transformer model first proposed the multi-head attention mechanism. This method is mainly used to solve sequence problems and can predict the output of different positions based on different focus points of the input. Compared with other attention mechanisms, the self-attention mechanism no longer relies on other network structures to learn to extract sequence features but directly obtains attention for the sequence itself, thus obtaining the desired attention features through the self-attention mechanism. The formula for the self-attention mechanism is shown in Equation (1):

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

Q, K, and V are feature matrices obtained by three matrix transformations of the input feature X, referred to as the query matrix, key matrix, and value matrix, respectively. In the self-attention mechanism, the input feature X is mapped to D-dimensional embedded feature X after embedding processing. Then, it is multiplied by three different weight matrices W_Q, W_K, and W_V, respectively, to obtain the query matrix Q, key matrix K, and value matrix V. Next, the similarity between Q and K is calculated to obtain the attention score, as shown in Equation 1, by taking the inner product of the Q vector of the current feature and the K vector of all features in the sequence. To ensure the gradient stability of Softmax and prevent excessive results, the attention score is scaled by dividing by $\sqrt{d_k}$ (where dk is the dimension of Q and K vectors). Then, the Softmax operation is used to normalize the attention score to obtain the attention weight. Finally, the attention weight is multiplied by the information matrix V to complete self-attention feature extraction and implement the self-attention mechanism.

The multi-head attention mechanism is another attention mechanism proposed at the same time as the Transformer model. It is based on the self-attention mechanism and performs a series of optimization operations. In the multi-head attention mechanism, multiple weight matrices W_Q, W_K, and W_V are trained to generate multiple sets of Q, K, and V feature matrices. The attention feature extraction method for each set of feature matrices is the same as that for single-head self-attention feature extraction calculation. After generating multiple sets of feature matrices, they are concatenated. When outputting the final result, the concatenated multi-head feature matrices are transformed into the final output features using the weight matrix W_O. Experiments have shown that different heads focus on different features, and the joint action of multiple heads can obtain better results. The formula for the multi-head self-attention mechanism is shown in Equation (2):

$$MultiHead(Q,K,V) = Concat(head_1,...,head_h)W^O \quad (2)$$

where is the aforementioned self-attention feature. The weight matrix dimensions are, h is the number of multi-head attention mechanisms, and d_{model} is the model dimension.

2.1.2. Depth Separable Convolution

Depthwise Separable Convolution refers to a convolutional method that combines depthwise convolution and pointwise convolution. Compared with traditional convolution, it can effectively reduce the number of network parameters and improve computational efficiency. In depthwise convolution, only the dependency relationship between sequences within each channel is considered, not between different channels. Pointwise convolution, on the other hand, focuses on the dependency relationship between different channels, not within the channel. Combining these two types of convolutions can achieve the effect of traditional convolution with fewer parameters.

2.1.3. Conformer Model Architecture

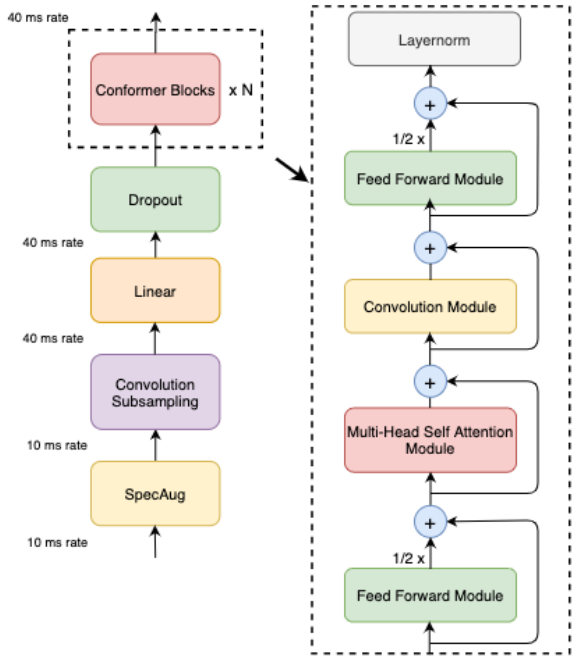

Figure 1: Conformer Model Architecture [13]

The overall architecture of the Conformer model is shown in Figure 1. First, it uses Convolution Subsampling to reduce the dimensionality of the input, which is then passed through a linear layer and dropout before being fed into multiple stacked Conformer blocks. Each Conformer block consists of three different modules: the Feed Forward Module, the Multi-Head Self Attention Module, and the Convolution Module. Residual connections are used between these three modules, with two Feed Forward Neural Network modules distributed at the beginning and end of the Conformer block. The outputs of both modules are multiplied by 1/2, then passed through layer normalization before being used as the output of the Conformer block. The formula for the Conformer block is shown in Equation (3). Finally, after passing through multiple Conformer blocks, the input obtains advanced acoustic features, which are then fed into the decoder for decoding, completing the entire process of the Conformer model. Using techniques such as depthwise separable convolution, residual connections, and multi-head self-attention, the Conformer model can achieve good performance.

Here, X represents the feature sequence input to the conformer block, b is the batch size, l is the length of the feature sequence, and d is the feature dimension. FFN refers to the Feed Forward Module, MHSA refers to the Multi-Head Self Attention Module, and Conv refers to the Convolution Module. x_i is obtained after passing through the four modules, residual connection, and layer normalization.

$$\begin{aligned}\tilde{x}_i &= x_i + \frac{1}{2} FFN(x_i) \\ x'_i &= \tilde{x}_i + MHSA(\tilde{x}_i) \\ x''_i &= x'_i + Conv(x'_i) \\ y_i &= Layernorm(x''_i + \frac{1}{2} FFN(x''_i))\end{aligned} \tag{3}$$

2.2. CTC Model

CTC is a loss function used for speech recognition, which can solve the problem of hard alignment. Its goal is to map the speech input sequence to the output label sequence. Since the length of the output label is shorter than that of the input speech frame, blank labels are inserted between repeated output labels to construct a CTC path with the same length as the input speech frame. Specifically, the speech input sequence of the CTC model is represented as X, the original output label sequence is represented as Y, and the CTC path is obtained from Y through the mapping function $B^{-1}(y)$. The encoder network transforms the acoustic feature x_t into a high-level representation h_{enct}, and then the CTC loss function is defined as the sum of negative log probabilities of all correct labels, As shown in Equations (4) and (5):

$$L_{CTC} = lnP(y \mid x) \tag{4}$$

$$P(y \mid x) = \sum_{q \in B^{-1}(y)} P(q \mid x) \tag{5}$$

where q is the CTC path. Through the conditional independence assumption, P(q|x) can be decomposed into a product of frame posteriors:

$$P(q \mid x) = \prod_{t=1}^{T} P(q_t \mid x) \tag{6}$$

where T is the length of the speech sequence.

3. Training Optimization

The training of end-to-end speech recognition requires a large amount of training data to achieve good performance. However, for low-resource languages, data scarcity is common, which can significantly affect the performance of the end-to-end models. Therefore, when training data is limited, improvements to end-to-end speech recognition techniques are necessary.

End-to-end models consist of a single network, and therefore the model parameters are updated through gradient descent without the involvement of human expertise. However, the computation of gradients is strongly influenced by the data and model structure, and thus model performance is often affected by the features and model structure. Although designing deeper and more complex networks can better explore the relationships between speech features, larger models also introduce more parameters, leading to poor training results. Blindly training large models does not seem to be suitable for end-to-end speech recognition when training data is limited.

The idea of transfer learning is to utilize knowledge learned from other domains to find similarities between two domains and apply knowledge from other domains to the target domain. Therefore, when training data is limited, directly using a model learned from the training data results in poor performance and can lead to overfitting or underfitting. Transfer learning can effectively use

knowledge learned from other domains to assist in model training, thereby preventing underfitting or overfitting problems and improving the performance of end-to-end speech recognition models.

In this article, a well-trained model from a resource-rich language is used as the pre-trained model for the low-resource language. During the training process of the low-resource language, the pre-trained model parameters of the high-resource language are first loaded as the initialization parameters of the low-resource language model. Then, the low-resource data is used to train the model, resulting in better speech recognition performance.

4. Experiment

4.1. Data

The XBMU-AMDO31 corpus is a Tibetan Amdo dialect speech corpus collected and recorded by Northwest Minzu University. It contains 31 hours of speech data from 66 speakers, including 32 males and 34 females. The speech data set is divided into training set, development set, and test set. The training set contains 18,539 sentences from 54 speakers, the development set contains 2,050 sentences from 6 speakers, and the test set contains 2,041 sentences from 6 speakers. Each speaker provided approximately 450 sentences, with a few individuals providing less than 200 sentences. All experiments in this paper were conducted using 80-dimensional log-Mel filterbank features, which were computed using a 25-millisecond window and shifted every 10 milliseconds. These features were normalized using speaker-dependent mean subtraction and variance normalization. At the current frame t, these features were stacked with the preceding 3 frames on the left and downsampled to a frame rate of 30 milliseconds.

4.2. Training

Table 1: Acoustic model settings

	Encoder	Decoder
Attention heads	4	4
Linear unit	2048	2048
Num blocks	12	6
CTC/Attention	0.3	0.3
Dropout rate	0.1	0.1
Input layer	Conv2d	/
Output size	256	/

The experiments in this article used a Mandarin speech recognition model trained on the Wenetspeech [15] dataset as the pre-training model, and were trained using the ESPnet [16] tool. In model training, the Adam [17] optimizer was used, and the attention loss and CTC loss were jointly trained [18] with a weight ratio of 0.7:0.3. Label smoothing (with a value of 0.1) was also used during training. The modeling units used in this article were Tibetan syllables. First, random initialization was used to train the XBMU-AMDO31 dataset, although the training loss performed well, the actual recognition results were not satisfactory. Then, the Wenetspeech pre-training model was used to initialize the model encoder and decoder parameters, and the Softmax layer was replaced with a Softmax layer suitable for Tibetan. As a result, the experimental results showed good performance. Table 1 shows the experimental parameters of the model in this article.

4.3. Results

The results of the baseline system were obtained based on the data released by XBMU-AMDO31,

which can be found in Table 2. As can be seen from the table, the model initialized with pretraining performs significantly better than the model initialized with random initialization. Transfer learning methods can help the target model leverage knowledge learned in other domains for training, thereby achieving better experimental results with limited training data.

Table 2: Results

System	Modeling Unit	Dev CER	Test CER
Baseline System	Syllable	14.8	13.8
Pretrained Model	Syllable	13.1	12.2

5. Conclusions

In this paper, we optimized the training process of Tibetan speech recognition and demonstrated the importance of transfer learning. Firstly, considering the correlation between languages, we chose Mandarin, which is a cognate language to Tibetan, as the source domain for transfer learning and used a pre-trained Mandarin model for initialization. This approach improved the performance of Tibetan speech recognition. Table 2 summarizes the comparison between this technique and the baseline system, proving the effectiveness of transfer learning in Tibetan speech recognition. Moreover, this method also provides insights for other low-resource speech recognition technologies.

Acknowledgments

This research was financially supported by a Basic scientific research business project of central universities (31920220010).

References

[1] Müller M. Dynamic time warping. Information retrieval for music and motion, 2007: 69-84.
[2] Juang B H, Rabiner L R. Hidden Markov models for speech recognition. Technometrics, 1991, 33 (3): 251-272.
[3] Pei C. Research on Tibetan Speech Recognition Technology Based on Standard Lhasa Tibetan [Doctoral dissertation, Tibet University].2009.
[4] Han Q., & Yu H. Research on Isolated Word Speech Recognition of Ando Tibetan based on HMM. Software Guide, 2010, 9 (7), 173-175.
[5] Zhao E., Wang C., Dang H., et al. Research on Isolated Word Speech Recognition Technology for Tibetan. Journal of Northwest Normal University (Natural Science Edition), 2015, 51 (5), 50-54.
[6] Zhang Y. Research on Lhasa Tibetan Speech Recognition Based on Deep Learning [Doctoral dissertation, Northwest Normal University]. Lanzhou, China. 2016.
[7] Graves A, Jaitly N. Towards end-to-end speech recognition with recurrent neural networks// International Conference on Machine Learning. JMLR. org, 2014.
[8] Graves A. Sequence transduction with recurrent neural networks. arXiv preprint arXiv: 1211. 3711, 2012.
[9] Chorowski J, Bahdanau D, Cho K, et al. End-to-end Continuous Speech Recognition using Attention-based Recurrent NN: First Results. Eprint Arxiv, 2014.
[10] Bahdanau D, Chorowski J, Serdyuk D, et al. End-to-end attention-based large vocabulary speech recognition//2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016: 4945-4949.
[11] Chan W, Jaitly N, Le Q, et al. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition//2016 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2016: 4960-4964.
[12] Lu L, Zhang X, Renais S. On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition//2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2016: 5060-5064.
[13] Gulati A, Qin J, Chiu C C, et al. Conformer: Convolution-augmented transformer for speech recognition. arXiv preprint arXiv:2005.08100, 2020.
[14] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need. Advances in neural information processing systems,

2017, 30.
[15] Zhang B, Lv H, Guo P, et al. Wenetspeech: A 10000+ hours multi-domain mandarin corpus for speech recognition//ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2022: 6182-6186.
[16] Watanabe S, Hori T, Karita S, et al. Espnet: End-to-end speech processing toolkit. arXiv preprint arXiv: 1804. 00015, 2018.
[17] Kingma D P, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv: 1412. 6980, 2014.
[18] Watanabe S, Hori T, Kim S, et al. Hybrid CTC/attention architecture for end-to-end speech recognition. IEEE Journal of Selected Topics in Signal Processing, 2017, 11 (8): 1240-1253.