An End-to-End Speech Recognition System Based on Shared Encoder

1st Zhengchang Wen
School of Software Engineering
South China University of Technology
Guangzhou, China
wzc3311@gmail.com

2nd Hailiang Huang
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou,China
sean@polyv.net

3rd Yingwei Liang
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou,China
alvinliang@polyv.net

4th Yi Ding
Guangzhou Easefun Co. Ltd.
Information and Technology
Guangzhou,China
dingyi@polyv.net

5th Xin Cheng
School of Software Engineering
South China University of Technology
Guangzhou, China
chengxin_19@aliyun.com

6th Qingyao Wu*
School of Software Engineering
South China University of Technology
Guangzhou, China
qyw@scut.edu.cn

***Abstract*—With the development of streaming media, automatic speech recognition (ASR) has been widely used in online education, live broadcast and other fields. However, for a better recognition effect in the real scenario, it is necessary to combine various technologies, such as front-end voice endpoint detection and back-end language model. In order to filter sensitive words in real scenarios, we require good online recognition and decoding methods. This paper presents an End-to-End speech recognition system, which unifies stream and non-stream speech recognition based on a shared encoder, and contains an additional CTC structure in the middle layer. Based on the monosyllable feature of mandarin, we calculate the probability distribution of syllables in the middle layer. The results show that our method is reliable for recognition in educational scenarios. We have achieved good results on aishell-1 and audio in real scenarios provided by the company. At the same time, this system provides accurate syllable information to analyze sensitive words further.**

***Index Terms*—automatic speech recognition, multi-task learning, shared encoder, multi modeling units**

I. Introduction

In recent years, streaming media has developed rapidly due to the stronger impact of sound and image. People have tended to complete interaction through audio and video. With the help of automatic speech recognition technology [1], users can transfer text information by speaking, which can also help us retain fleeting information. Similarly, with the help of the converted text, key or illegal content in the video can be retrieved more quickly without watching the lengthy video. The speech recognition system has been applied in various fields, such as online education, conference, live broadcast, chat software, etc.

In recent years, due to the stronger impact of sound and image, streaming media has developed rapidly. People have tended to complete interaction through audio and video. With the help of automatic speech recognition technology[1], users can transfer text information by speaking, which can also help us retain fleeting information. Similarly, with the help of the converted text, key or illegal content in the video can be retrieved more quickly without watching the lengthy video. Nowadays, speech recognition system has been applied in various fields, such as online education, conference, live broadcast, chat software and so on.

In many cases, we need an online recognition system that can write something while listening. Sometimes, all we need is to be able to convert our speech fragments into words offline. However, online speech recognition algorithms face the disadvantages of low accuracy and relying on language models like N-gram [2][3][4]. Although the offline speech recognition algorithm has a high accuracy rate, it can not achieve the convenient writing function when listening. Therefore, in many scenarios, we often need to train two sets of speech recognition algorithms, which significantly increases the computational cost. At the same time, in some scenarios, sensitive words need to be quickly identified and shielded, and the model needs to have accurate real-time recognition ability for these words.

In many cases, what we need is an online recogni-

tion system that can write something while listening. Sometimes, all we need is to be able to convert our speech fragments into words offline. However, on-line speech recognition algorithms face the disadvantages of low accuracy and relying on language models like N-gram[2][3][4]. Although the offline speech recognition algorithm has a high accuracy rate, it cannot achieve the convenient function of writing when listening. Therefore, in many scenarios, we often need to train two sets of speech recognition algorithms, which greatly increases the computational cost. At the same time, in some scenarios, sensitive words need to be quickly identified and shielded, and the model needs to have accurate real-time recognition ability for these words.

Online speech recognition algorithms are mainly divided into CTC [5][6], RNN-T[7] and other algorithms. CTC algorithm makes the assumption of time independence and completes the alignment of unequal length acoustic feature sequences and text sequences. It is the basis of stream model. However, due to the assumption of independence, its ability to capture the previous and subsequent information is not strong, and it cannot learn the text information. The RNN-T uses an additional RNN[8] network structure to ensure that the text content of the previous text can be input into the following text. However, the training of the RNN-T is time-consuming. The commonly used method of non-stream model is based on attention[9] . Due to the operation mechanism of attention, the model can often learn the relevant information of the previous and subsequent text, and then outperform the online method in accuracy. However, despite the existence of position encoding, the attention-based method[10] often ignores the position information[11], and requires more data and deeper networks. For the screening of sensitive words, the common scheme is to take a certain delay to detect the text content of a sentence after complete recognition. Another scheme is to match the sound segments, that is, record some audio, and then score the similarity. The disadvantage of this method is that it is difficult to collect audio signals and cannot meet the changing business needs in actual production.

In order to improve the above problems, we designed a speech recognition system. We use a strategy of sharing weights, combining the encoder of stream model and non-stream model, and adopt the idea of multi task learning to share the encoder of CTC decoder and attention-based decoder. The monotonicity of CTC decoder can help the attention-based decoder learn more location information. Considering that CTC can converge rapidly, and the attention mechanism needs a deeper network structure, we also designed multi CTC learning tasks. We consider the loss of outputting an additional CTC in the middle layer of the encoder.We calculate the probability distribution of its syllables.Because mandarin syllables actually do not need much context information, such a structure can enable the deeper network to learn more advanced information. At the same time, this structure can also enable the algorithm to generate more accurate syllable information, thus assisting the sensitive word matching module.

II. Related work

Speech recognition algorithms are mainly divided into traditional HMM[12][13][14] or its combination with deep learning methods[15], and the end-to-end neural network architecture that has become popular in recent years. Because of its simple model architecture: an encoder extracts acoustic features and a decoder converts them into text sequences, this model architecture provides a lot of design space. Similarly, the end-to-end model is divided into two architectures. One is a stream model such as CTC and RNN-T, and the other is a non-stream model based on attention[16][17]. The advantage of CTC is the fast decoding speed, while the advantage of the attention-based model is that it can consider the global information, and its common point is that they both use encoder.

In recent years, a model framework, CTC/attention hybrid architecture[18][19], has become increasingly popular. It has proved that combining the reasoning results of the two, or combining the learning of the two[20], is compelling. In view of the low efficiency of the CTC model, a common method is to combine language models to improve it. The disadvantages of the attention-based model, such as transformer, are that it cannot capture the alignment of acoustic features and text and lacks the reading of location information. The solution in the industry usually uses the network architecture of the conformer[21]. Conformer combines the advantages of CNN[22][23] and transformer, which is better at capturing local[11] information. Convolution can ensure that the input acoustic features will not be disturbed in the time dimension.

As for modeling units, different languages usually have different ways. There are phonemes, grapheme, symbols, words, sub words, and characters[24][25][26]. In recent years, the use of grapheme, which is the smallest unit of language writing, has

gradually occupied a place. This selection method does not need the help of phonologists to formulate complex professional lexicons. However,for mandarin, syllable is a reliable way[27]. First, since mandarin is a monosyllabic language, and one syllable corresponds to one mandarin character, it is easy to get the labeled data. Moreover, the syllable is a discrimination form of sound transmitted to the human ear, so it is also a relatively reliable modeling unit.

III. Proposed Methods

We want a unified model with strong robustness, which can use the advantages of both the stream model and non-stream models. Therefore, we consider sharing the two. In the encoder part of the shallow layer, we calculate the probability distribution of syllables and calculate CTC loss between layers. In terms of overall architecture, we adopt the architecture of transformer and CTC and use convolution layers for subsampling.We suppose $X = (x_1, x_2, x_3, ..., x_t)$ is the input acoustic feature,$Y = (y_1, y_2, y_3, ..., y_n$is the text sequence,and $Z = (z_1, z_2, z_3, ..., z_k)$is the syllable sequence.

A. Overall process

As shown in the Fig.1, this is the overall architecture of our network. From left to right, we first perform time-based convolution on the input sequence to extract key features and perform subsampling. After passing through the linear layer and position encoding, it is input into the encoder. The output of the encoder is fed into the linear layer to map the attention dimension to the vocabulary dimension, and together with the text label to calculate the final CTC loss. The encoder is composed of multiple blocks, and the structure of each block is the same. For the architecture of the attention decoder, we adopt the transformer as our attention decoder.

Conformer encoder:The conformer encoder is a model architecture that combines the advantages of transformer and CNN. Because the transformer can not learn local information well, the conformer encoder introduces convolution based on attention. The conformer encoder comprises four modules. The first feedforward module including macaron structure, multi-head self-attention module, convolution module that can learn local information, and the second feedforward module.

CTC decoder:The CTC part makes a time independence assumption. The output of CTC is the probability distribution of a modeling unit dimension.In the prediction stage, for each given input X, CTC needs to find the output y corresponding to the maximum probability.It can be expressed as follows:

$$Y^* = \arg_Y \max P(Y|X) \tag{1}$$

The CTC does not need to input and output alignment. However,for a given input, in order to calculate the probability corresponding to Y, it is still necessary to sum all possible aligned probabilities because there may be multiple output paths corresponding to the same output. The CTC uses the empty set ϕ to assist in calculating alignment. For example,$\phi\phi c\phi a\phi t$ and $\phi c\phi\phi a\phi t\phi$ corresponds to the probability distribution of "cat". After the CTC statistics are aligned, we calculate the posterior probability of "cat". This probability can be described as follows:

$$P(Y|X) = \sum_{F(l)=y} \prod_{t=1}^{T} z_{l_t}^t \tag{2}$$

where l_t represents the output character corresponding to the path l at time step T, and $z_{l_t}^t$ represents the probability that the output is l_t at time t.We want to maximize Y in the case of X. For backpropagation, the optimization objective is to minimize the negative likelihood of $P(Y|X)$ as follows where D presents the training set:

$$L_{ctc_{ch}} = \sum_{X,Y \in D} -\log(P(Y|X)) \tag{3}$$

B. Shared encoder and syllable modelling units

Everyone Chinese learned to speak from Pinyin when he was a child. He learns Pinyin first, then speaks, and then writes. Similarly, we hope the network will do the same. As mentioned above, CTC lacks the ability to understand the language, and learning language laws depends entirely on the paired texts we annotate. However, for mandarin, syllable is an essential language level knowledge. At the same time, we hope to help the network learn more with the help of the shared encoder. The CTC-based model has monotonicity, while the attention-based model can better learn the global content. So we used a shared encoder, which can combine both characteristics. At the same time, considering the characteristics of previous CTC models, we think that it does not need deeper layers, so in the middle layer, we calculate CTC based on syllable loss. Here we use Pinyin as syllable units just like we show on syllable unit block on Fig.1. After that, we get CTC loss based on syllable:

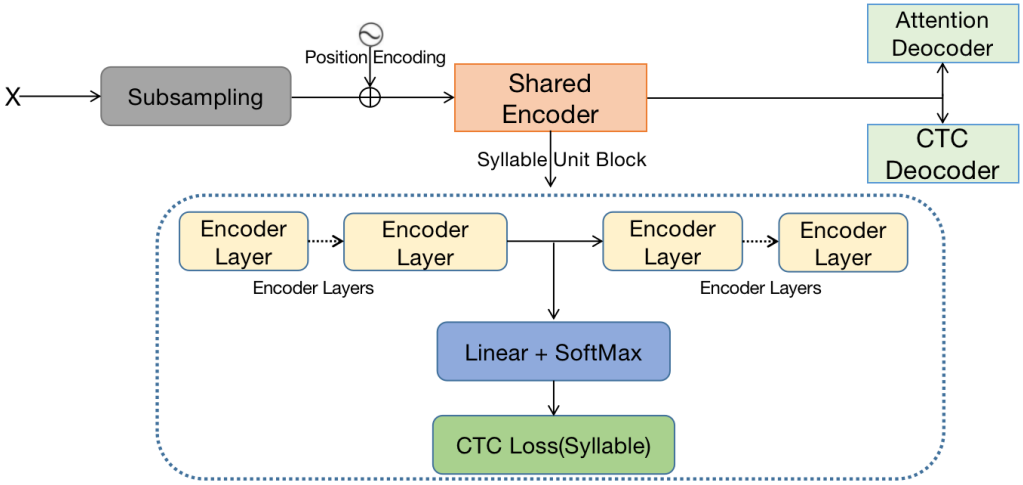

Fig. 1. Shared encoder with a syllable unit block.The syllable unit block represents our proposed modeling unit module according to syllables. Such an architecture can be added between layers.

$$L_{ctc_{sy}} = \sum_{X,Z \in D} -\log(P(Z|X)) \tag{4}$$

Z here refers to all possible syllable sequences. Next, after similar encoder layers. During training, the model will eat the paired speech and text and simultaneously calculate the CTC loss and attention loss. In backpropagation, we combine them by linear combination:

$$L_{ctc} = \lambda L_{ctc_{sy}} + (1-\lambda) L_{ctc_{ch}} \tag{5}$$

L_{ctc} here refers to the loss function of CTC, where D is also from our training set. L_{ctc} is the loss of CTC part we consider. For text and syllables, we do not need to collect additional data. We only need a mapping transformation to convert the original text data into syllable data. We can also calculate the loss function based on attention and combine them through linear combination:

$$L = \zeta L_{ctc} + (1-\zeta) L_{att} \tag{6}$$

In the above two equations,ζ and λ are hyperparameters we set. When reasoning, we refer to the output of two decoders in parallel. After the acoustic feature input, results are generated by the CTC encoder. After a sentence is finished, we can refer to the probability distribution generated by the attention-based decoder. Finally, we can use the double scoring method to get the results of our speech recognition. Similarly, the generated syllable results can also be embedded or connected to the language model. Thus, a speech recognition system with strong expansibility and complete functions is constructed.

IV. Experiment and Visualization

A. Enviroment

The model was trained on a Linux server running Ubuntu 20.04 LTS (GNU/Linux 5.4.0-122-generic x86_64), and we used Python to implement the algorithm. The code environment is Python 3.8, CUDA version 10.2, and torch version 1.12. After training the model, we test the model through the GPU and CPU.

B. Dataset

We used aishell-1[28] as our basic data set, which contains 178 hours of data collected from 400 people. Easefun provides us with follow-up tests and some training data. As for syllables, we use the open-source text-to-syllable unit tool to help us obtain labeled syllable data.

C. Setup

According to the sliding window with the size of 25ms, all the acoustic features we input each time are 80-dimensional filter banks calculated based on the length of 10ms time. The vocabulary size is 4231 Chinese words and 1352 Pinyin vocabulary combinations with four tones. We set the λ in Eq.(5) and ζ in Eq.(6) to 0.3. There are 12 layers of encoder layer. The dimension of attention is 256, and we use the relative position coding.We calculate the CTC loss of syllables in the third encoder layer. The convolution

kernel in the transformer is 15 dimensions. We use a 6-layer transformer for the decoder. During the training, we used Adam to help us with gradient descent, with a learning rate of 2e-3.After that, we test our model by the weighted sum of the CTC and attention scores, where the CTC beam size is set to 10.

D. Results

We use CER, the character error rate, as the evaluation index. The results are shown in Table 1.

TABLE I
COMPARISON OF CERS OF DIFFRENT MODELS ON AISHELL-1

Model	Dev	Test
Conformer	5.1	5.4
Transformer	5.5	5.9
Conformer + Syllable Units	5.0	5.2
Transformer + Syllable Units	5.3	5.5

The experimental results show that the model with shared conformer encoder achieves a CER of 5.4% on aishell-1. After adding syllable-based modeling units, the CER of the model can reach 5.2%.It can be seen that conformer can achieve better results than transformer, and the syllable based modeling method we proposed can further enhance the recognition effect. Under the real scene data provided by Easefun, the CER of our model can reach 7.1%. Due to our syllable modeling unit, our model can accurately identify the syllable information about Mandarin. At the same time, we introduce a vocabulary of sensitive words and use it in real scenarios.

After training, due to the limited computing resources, we need to migrate the model to the client, and we will recognize and mask the incoming streaming voice in real-time. The steps are as follows: First, the stream recognition result is returned to the sensitive word screening module, then the sensitive word screening module performs real-time analysis according to the context and syllable, and then returns the real-time screened text. The voice signal on the calculation timestamp is masked according to the text, and the corresponding text is not returned. It can be proved that our model can achieve good results with low latency in about 5 seconds.

V. CONCLUSION

In this paper, we use a shared encoder based on stream and non stream models. It provides accurate syllable information and improves the overall recognition effect. Finally, we deployed and tested the model on the server side and the client side. Our model provides high robustness and scalability in real scenarios. In future work, we will further explore the role of syllables in speech recognition and their combination with language models.

VI. CONCLUSION

In this paper, we use a shared encoder based on stream and non stream models,and combine these two models' advantages. As for how to share, in the shared encoder, based on the principle of mandarin pronunciation and the process of human language learning, we propose to calculate the probability distribution based on syllables in the middle encoder layer. It provides accurate syllable information and improves the overall recognition effect. Finally, we deployed and tested the model on the server side and the client side. Our model provides high robustness and scalability in real scenarios. In future work, we will further explore the role of syllables in speech recognition and their combination with language models.

ACKNOWLEDGMENT

This work was supported by National Natural Science Foundation of China (NSFC) 61876208, Tip-top Scientific and Technical Innovative Youth Talents of Guangdong Special Support Program (2019TQ05X200) and 2022 Tencent Wechat Rhino-Bird Focused Research Program Research (Tencent WeChat RBFR2022008).

REFERENCES

[1] K. F. Lee. "Automatic Speech Recognition: The Development of the SPHINX System". In: *Aston University* (1989).

[2] W. B. Cavnar and J. M. Trenkle. "N-Gram-Based Text Categorization". In: (2001).

[3] A. Pauls and K. Dan. "Faster and Smaller N-Gram Language Models". In: *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*. 2012.

[4] Chengxiang Zhai and John Lafferty. "A Study of Smoothing Methods for Language Models Applied to Information Retrieval". In: *Acm Transactions on Information Systems* 22.2 (2004), p.179–214.

[5] A. Graves, S Fernández, and F. Gomez. “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks”. In: *International Conference on Machine Learning*. 2006.

[6] A. Das et al. “Advancing Connectionist Temporal Classification With Attention Modeling”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018.

[7] A. Graves. “Sequence Transduction with Recurrent Neural Networks”. In: *Computer Science* 58.3 (2012), pp. 235–242.

[8] W. Zaremba, I. Sutskever, and O. Vinyals. “Recurrent Neural Network Regularization”. In: *Eprint Arxiv* (2014).

[9] A. Vaswani et al. “Attention Is All You Need”. In: *arXiv*. 2017.

[10] W. Chan et al. “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition”. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016.

[11] B. Yang et al. “Modeling Localness for Self-Attention Networks”. In: 2018.

[12] L. R. Rabiner. “Juang: An introduction to hidden Markov Models”. In: ().

[13] Lawrence R. Rabiner. “A tutorial on hidden Markov models and selected applications in speech recognition”. In: *Proc IEEE* 77 (1989).

[14] B Schuster-Böckler and A. Bateman. “An introduction to hidden Markov models”. In: *Curr Protoc Bioinformatics* (2007).

[15] G. Saon and J. T. Chien. *Large-Vocabulary Continuous Speech Recognition Systems*. Fundamentals of speech recognition /, 2012.

[16] D. Bahdanau, K. Cho, and Y. Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate”. In: *Computer Science* (2014).

[17] Y. Kim et al. “Structured Attention Networks”. In: (2017).

[18] Christoph Wick, Jochen Zllner, and Tobias Grüning. “Rescoring Sequence-to-Sequence Models for Text Line Recognition with CTC-Prefixes”. In: *International Workshop on Document Analysis Systems*. 2022.

[19] N. Jung, G. Kim, and H. G. Kim. “Back from the future: bidirectional CTC decoding using future information in speech recognition”. In: (2021).

[20] G. Chen et al. “GigaSpeech: An Evolving, Multi-domain ASR Corpus with 10,000 Hours of Transcribed Audio”. In: *arXiv e-prints* (2021).

[21] A. Gulati et al. “Conformer: Convolution-augmented Transformer for Speech Recognition”. In: 2020.

[22] A. Krizhevsky, I. Sutskever, and G. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Advances in neural information processing systems* 25.2 (2012).

[23] T. N. Sainath et al. “Improvements to deep convolutional neural networks for LVCSR”. In: *arXiv e-prints* (2013).

[24] S. Zhang et al. “Investigation of Modeling Units for Mandarin Speech Recognition Using Dfsmn-ctc-smbr”. In: *ICASSP2019*. 2019.

[25] W. Zou et al. “Comparable Study Of Modeling Units For End-To-End Mandarin Speech Recognition”. In: *2018 11th International Symposium on Chinese Spoken Language Processing (ISCSLP)*. 2018.

[26] J. Kim and P. Kang. “K-Wav2vec 2.0: Automatic Speech Recognition based on Joint Decoding of Graphemes and Syllables.” In: (2021).

[27] X. Wang et al. “Cascade RNN-Transducer: Syllable Based Streaming On-device Mandarin Speech Recognition with a Syllable-to-Character Converter”. In: (2020).

[28] H. Bu et al. “AISHELL-1: An open-source Mandarin speech corpus and a speech recognition baseline”. In: *Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases Speech I/o Systems Assessment*. 2017.