INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 05 Issue 05 May 2022

DOI: 10.47191/ijmra/v5-i5-12, Impact Factor: 6.261

Page No. 965-970

A Study of Speech Recognition with Deep Learning

Feng Li[1], Yiyang Wei[2]

[1,2]School of management science and Engineering, Anhui University of Finance and Economics, Bengbu 233030, China

ABSTRACT: The development of deep learning and the continuous progress of artificial intelligence have contributed to the rapid development of speech recognition. Among them, the end-to-end structure is the more important part of the whole speech recognition. This paper introduces two end-to-end speech recognition methods, the attention model and the CTC loss function, describes the practical application of deep learning in speech recognition and suggests improvements to the two models. Finally, the practical usefulness of speech recognition is demonstrated by analyzing the application of trigger word detection and sentiment analysis in artificial intelligence in teaching and learning.

KEYWORDS: Speech Recognition; Deep learning; CTC Loss Function; Sentiment Analysis.

I. INTRODUCTION

Speech recognition is a technology that uses machines to recognize and understand speech signals and convert them into corresponding text and commands [1]. In simple terms, speech recognition can find the corresponding text when given a sound segment. The two most important parts of speech recognition are acoustic feature extraction and acoustic model building. One of the most used acoustic features is the Mel Frequency Cepstrum Coefficient (MFCC) [2]. For acoustic model building, the Gaussian Mixture Model (GMM) [3] in the traditional GMM-HMM [4] acoustic model ignores temporal information and does not make sufficient use of before and after information, which has certain limitations. The rise of deep learning, on the other hand, provides a new approach to acoustic modeling. Deep convolutional neural networks can improve the connectivity between frames, i.e. the closeness of the before and after information, and can achieve an improved level of speech recognition in the process of speech recognition. However, the emergence of accurate speech recognition was marked by the development of the sequence-to-sequence model [5]. Microphones work by measuring small changes in air pressure [6], and we can now hear people speaking because our ears detect small changes in air pressure generated by speakers or headphones. Speech recognition algorithms, which use a segment of an audio graph (an image of air pressure against time) as input, are then fed with the corresponding text. Even the physiological structures in the human ear are able to measure the intensity of different frequencies, rather than processing the original sound waves in their raw form [7]. So a common pre-processing step for sound frequency data is to generate a spectrogram from the sound frequency fragments, inside this graph the horizontal coordinates are time and the vertical coordinates are frequency, with different shades of color indicating energy levels. The spectrogram indicates how much volume is present at different times and at different frequencies.

Speech recognition systems were constructed based on phonemes (the basic units of sound) [8], where researchers divided language into basic sound units, and linguists believed that representing audio in terms of phonemes was the best way to recognize speech. Later, scholars proposed end-to-end structures, which could solve the situation of sequence misalignment during speech recognition [9]. Moreover, there is no need to use phonemes to represent sounds. However, the implementation presupposes that a larger dataset is required. An academic dataset for speech recognition may b 300 hours long, and in the academic community, an audio dataset of 3000 hours would be considered a reasonable size. This paper presents two types of end-to-end speech recognition based on attention models and CTC loss functions respectively. Of these, the CTC loss function is the main core algorithm for the end-to-end architecture [10].

II. PREVIOUS STUDIES

(1) Speech recognition based on attention models

Attentional modeling is a technique for extracting effective features from a sequence of features in a real sequence-to-sequence model. When people observe a scene, they pay different attention to different locations, people, and things within that scene.

Derived from this phenomenon, attention models calculate a series of attention weights [11]. In speech recognition, a soft attention model is generally used, i.e. the attention weights are calculated for all encoder output data. A speech recognition based on the attention model is shown in the Figure 1.

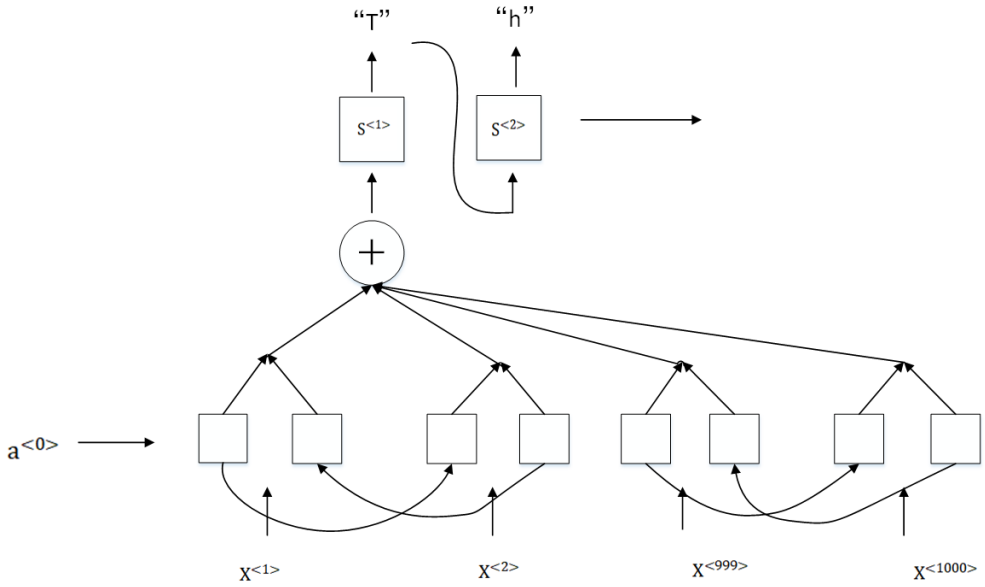

Figure 1 Diagram of speech recognition based on the attention model

The model consists of an encoder, a decoder, and an attention layer. The encoder is represented using a bidirectional RNN, while the decoder is composed using a unidirectional RNN [12]. First, the input data is fed through the bidirectional RNN model, which $\vec{a}^{<t>}$ denotes the time step t in the forward propagation of the activation, and $\overleftarrow{a}^{<t'>}$ denotes the time step t in the backward propagation of the activation.$a^{<t'>}$ Connecting these two activation units, their representation of the feature vector at time step t

$$a^{<t'>} = (\vec{a}^{<t'>}, \overleftarrow{a}^{<t'>})$$

The input to a unidirectional RNN depends on the attention layer. The attention layer has an attention sub-network, which contains only one implicit layer and can be $e^{<t,t'>}$ denoted by the sub-network as shown in the following figure.

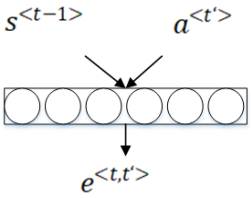

Figure 2 Attentional sub-network

where the first input $s^{<t-1>}$ is the previous step in the neural network decoded at the previous moment and $a^{<t'>}$ is the other input. In simple terms, the amount of attention needed for the input depends mainly on the activation from the previous state. By training this small neural network and back-propagating the algorithm to find the corresponding function utilizing gradient descent.

Then, all moments thee $e^{<t,t'>}$ are exponentially normalized. The normalized value is the attention weight [13], which can be interpreted as the attention of the output to the input. For example, when the decoder outputs a character, it should pay attention to how much attention is paid to thet′ input character is how much. This is shown below.

$$a^{<t,t'>} = \frac{\exp(e^{<t,t'>})}{\sum_{t'=1}^{N} \exp(e^{<t,t'>})}$$

Finally, the features are weighted and summed over all moments to obtain the output corresponding to the position of the output sequence under the attention model o.

$$o^{<t>} = \sum_{t'} a^{<t,t'>} a^{<t'>}$$

(2) Speech recognition based on CTC loss function

In the horizontal coordinates, the audio is input at various times and the text is output by the attention model, which is a good approach, but the algorithm is prone to run at a quadratic cost. A better alternative approach to speech recognition is to use the CTC loss function, which stands for Conjointist Temporal Classification [14]. The principle is described in the figure.

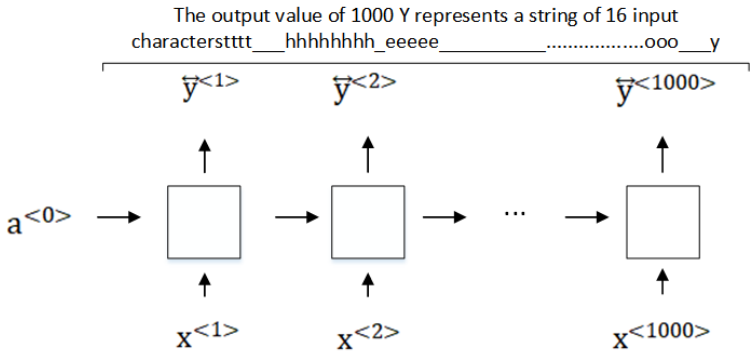

Figure 3 Diagram of speech recognition based on CTC loss function

If there is an audio clip "The handsome boy", a simple one-way RNN is used here, but in practice, this is usually a bi-directional LSTM or bi-directional GRU, usually a deeper model. The important thing to note here is that the times here are large and that in speech recognition the input times usually greatly exceed the output times. For example, if you have 5 seconds of video with features at 50Hz, i.e. 50 samples of data per second, then 5 seconds of audio will be 250 input characters.

$$50\text{Hz} * 5\text{s} = 250$$

The basic principle of the CTC loss function is to collapse repetitive characters that are not split by whitespace, for example, an RNN produces an output sequence "bbbbbbb ____ooo_y", which is the correct output for "boy", using underscores to represent a special whitespace character for clarity. a character many times, giving a sequence of 1000 outputs. So by inserting a bunch of whitespace characters, you can still end up with a shorter string of text, so the paragraph above actually has 16 characters, and if somehow it can use the output value of these 1000 Y's to represent a string of these 16 characters.

Both attention-based models and CTC models are viable solutions for speech recognition. Today, building speech recognition systems still require significant effort and data sets.

III. APPLICATION AREAS

(1) Trigger word detection in AI

There is a very important branch in the field of artificial intelligence, natural language processing, which simply means enabling machines to understand human language and communicate with humans. Natural language processing covers machine translation, intelligent retrieval, and more. With the increasing maturity of deep learning, it simplifies the models in natural language processing but improves performance, trigger word detection being an example. As voice recognition is used in more and more smart devices, people can use their voice to command the devices they own, which is known as activating this detection [15].

Examples of activated word detection include Amazon Echo using "Alexa" to wake up, Baidu DuerOS using "Hello Xiaodu", Apple Siri using "Hey Siri" and Google Home using "ok Google". "For example, when a user says "Hello Xiaodu, what time is it? Baidu DuerOS will wake up with "Xiaodu Hello" and answer the corresponding voice question. So, if you can create an activation vocabulary detection system, then it is possible that you can activate your computer to make it do what you want. Activated vocabulary detection is still being adapted, so there is not a single universally default optimal algorithm, so it is described below using RNN. The principle of activated vocabulary detection is shown in the diagram.

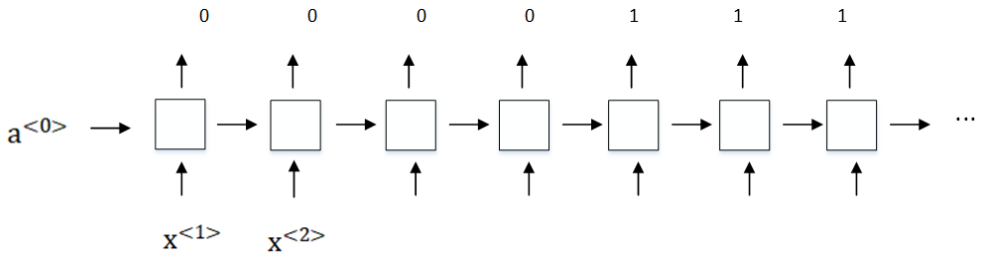

Figure 4 Schematic diagram of trigger word detection

Taking "Hey Siri" as an example, when "Siri" is heard, the previous target tag in the training set can be set to 0, and the next target tag after it is detected can be set to 1. And so on, when the trigger word is responded to, the next subsequent target tag is set to 1, and the previous target tags are all 0. However, this RNN-based trigger word detection is flawed and can lead to an unbalanced training set with far more zeros than ones.

In terms of improvement, there are two methods. The first method is to set multiple subsequent target tags to 1 relative to a single target tag after a trigger word has been detected. The other method is to set the target tag value to 1 for a certain period after detection. The improvement allows for a balanced ratio of 0s to 1s.

3.2 Affective analysis in teaching

The task of sentiment analysis is to analyze a piece of text and tell people whether someone likes what they are discussing or not. It is the most important component of natural language processing and is used in many applications. There is this sentiment analysis problem. Input a piece of text and the output can be the sentiment that you want to predict. Examples are the emotions such as happiness, sadness, anger, boredom, and fear that students produce in the classroom. Sentiment analysis can use a piece of text to predict whether the students' feelings towards the teacher in class are positive or negative, and then the teacher and school can see if the teacher is having problems or whether the students' attitudes towards the teacher have changed for the better or worse over time. One of the challenges with sentiment analysis is the lack of a particularly large labeled training set in [16], but with the use of word embeddings, it is possible to build a good sentiment analyzer relying on a moderately sized labeled training set in [17]. For sentiment analysis tasks, it is common that the training set may have between 10,000 and 100,000 words of data. Sometimes it is even less than 10,000 words, and word embeddings can help to understand what happens when the training set is small.

(1) Word embedding method

Suppose you have a sentiment analysis model like this. You can take a sentence like "The teacher is excellent" and look up these words in your dictionary. This is shown in the diagram.

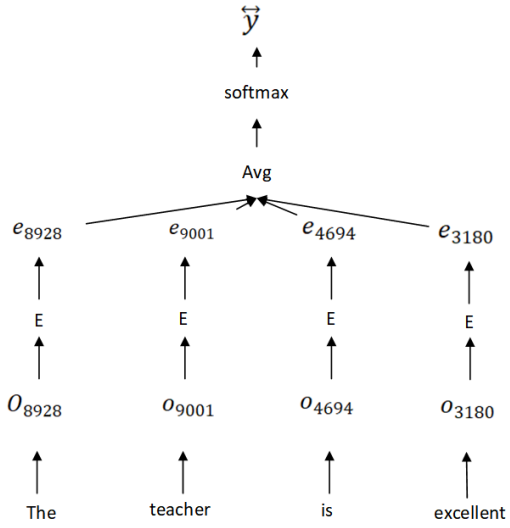

Figure 5 Schematic diagram of word embedded sentiment analysis

This sentence has a total of four terms. First, we get "The {0, 8, 9, 2, 8}", which is the product of the one-hot vector and the embedding matrix E. It can continue to learn into larger text vectors. It can continue to learn from a larger vector of text. This vector is then used to extract the embedding vector for the word "the"e_8928. The same operation is then performed for the remaining three words. Finally, the classifier is constructed using the averaging method, averaged over them, and then transferred to the softmax classifier, which outputs the predictions [18].

Although the averaging method gives a better prediction, what it does is average the meanings of all the words in the example. One of the problems with this algorithm is that it ignores the order of the words, particularly, in the case of the following comment, "Completely lacking in good teaching methods, good passion, and good attitude", although this is a negative comment. However, it has multiple occurrences of "good", so if you use the average algorithm output, this ignores the order in which the words are arranged and simply averages all the words. If there are many positive representations in the feature vector. The classifier may simply assume that this is a good rating, which in turn affects the analysis of sentiment, however in reality it is a very negative review.

(2) Deep learning approach

There is more sophisticated model which not only sums all word embeddings but also performs sentiment analysis using an RNN. the previous steps are similar, find the one-hot vector for each word in the comment, then multiply the word embedding matrix E. You can then use the resulting embedding vectors of multiple words as input to the RNN, and finally predict the outcome through a many-to-one RNN model. This is shown in Figure 6.

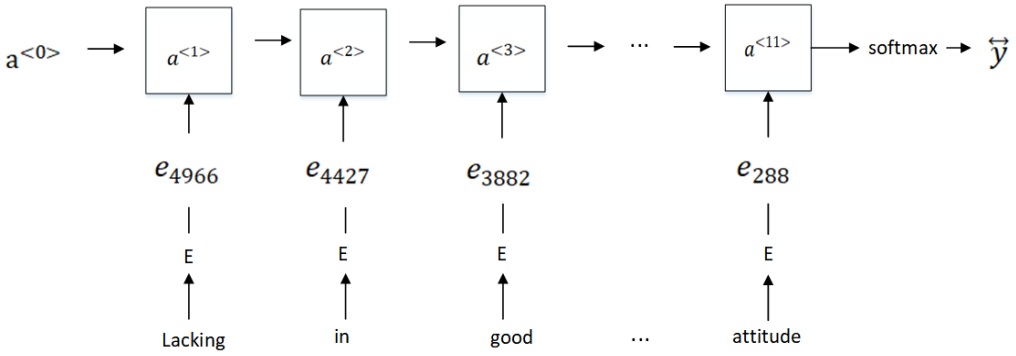

Figure 6 Schematic diagram of deep learning-based sentiment analysis

Like word embedding, the one-hot vector for each word is found and, as usual, multiplied with the word embedding matrix E. Multiple words embedding vectors can then be obtained. These word embedding vectors are used as input to the RNN, which finally computes the predicted sentiment results in the final step. With an algorithm like this, the order of the words can be considered to get better results and to realize that this comment is negative [19]. But the nature of the algorithm differs from the previous algorithm in that it recognizes "not good" as a negative comment. The previous algorithm just added everything into a larger word vector and did not realize that the meaning of "not good" was different from where "good" was used.

A larger dataset could be used to train the word embedding vector and better results obtained. This can even be generalized to words that do not appear in the training set. For example, enter the following text "Completely absent of good teaching methods", even though "absent" is not in your label training set, if it is in the billion-word or 100-billion-word database of the training word embedding vector of billions or hundreds of billions of words, you may be able to get the correct result, or even better generalize to words that are present in the training set of the training word embedding vector, but not necessarily in the label training set. This label training set may have been prepared by the researchers specifically for the sentiment analysis problem.

IV. CONCLUSIONS

This paper focuses on the principles of the attention model and the CTC loss function on the one hand, and the application of RNNs in speech recognition on the other. The end-to-end model based on the attention model does not require a priori alignment information, nor does it require independence tests between phoneme sequences, nor does it require artificial methods such as pronunciation dictionaries, and can be effectively combined with neural networks to achieve speech recognition. CNN, Transducer, and other models for improvement and optimization. On the other hand, two applications of speech recognition under deep learning are introduced. Trigger word detection and sentiment analysis, as promising fields, have been developed for a long time.

ACKNOWLEDGMENT

This work was supported in part by the Natural Science Foundation of the Higher Education Institutions of Anhui Province under Grant No. KJ2020A0011, Innovation Support Program for Returned Overseas Students in Anhui Province under Grant No. 2021LCX032, Undergraduate teaching quality and teaching reform project of Anhui University of Finance and Economics under Grant No. acszjyyb2021035, Undergraduate Research and Innovation Fund project of Anhui University of Finance and Economics under Grant No. XSKY22154.

REFERENCES

1) P. Wang, "Research and Design of Smart Home Speech Recognition System Based on Deep Learning," 2020 International Conference on Computer Vision, Image and Deep Learning (CVIDL), 2020, pp. 218-221.
2) A. Winursito, R. Hidayat and A. Bejo, "Improvement of MFCC feature extraction accuracy using PCA in Indonesian speech recognition," 2018 International Conference on Information and Communications Technology (ICOIACT), 2018, pp. 379-383.

3) T. R. Kumar, S. Padmapriya, V. T. Bai, P. M. Beulah Devamalar and G. R. Suresh, "Conversion of non-audible murmur to normal speech through Wi-Fi transceiver for speech recognition based on GMM model," 2015 2nd International Conference on Electronics and Communication Systems (ICECS), 2015, pp. 802-808.
4) J. Rahman Saurav, S. Amin, S. Kibria and M. Shahidur Rahman, "Bangla Speech Recognition for Voice Search," 2018 International Conference on Bangla Speech and Language Processing (ICBSLP), 2018, pp. 1-4.
5) M. Mimura, S. Ueno, H. Inaguma, S. Sakai and T. Kawahara, "Leveraging Sequence-to-Sequence Speech Synthesis for Enhancing Acoustic-to-Word Speech Recognition," 2018 IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 477-484.
6) F. Mitsugi, S. Kusumegi, T. Kawasaki, T. Nakamiya and Y. Sonoda, "Detection of Pressure Waves Emitted From Plasma Jets With Fibered Optical Wave Microphone in Gas and Liquid Phases," in IEEE Transactions on Plasma Science, vol. 44, no. 12, pp. 3077-3082, Dec. 2016.
7) Liu Chien Chih and Chiang Che Ming, "The effect of environment of different noise frequencies on human physiological responses," 2011 International Conference on Multimedia Technology, 2011, pp. 1808-1811.
8) N. Uma Maheswari, A. P. Kabilan and R. Venkatesh, "Speaker independent speech recognition system based on phoneme identification," 2008 International Conference on Computing, Communication and Networking, 2008, pp. 1-6.
9) C. Fan, J. Yi, J. Tao, Z. Tian, B. Liu and Z. Wen, "Gated Recurrent Fusion With Joint Training Framework for Robust End-to-End Speech Recognition," in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 29, pp. 198-209, 2021.
10) J. Sun, G. Zhou, H. Yang and M. Wang, "End-to-end Tibetan Ando dialect speech recognition based on hybrid CTC/attention architecture," 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2019, pp. 628-632.
11) J. -n. Chen, S. Gao, H. -z. Sun, X. -h. Liu, Z. -n. Wang and Y. Zheng, "An End-to-end Speech Recognition Algorithm based on Attention Mechanism," 2020 39th Chinese Control Conference (CCC), 2020, pp. 2935-2940.
12) H. Zhang, "An Exploration of Recurrent Units for Automatic Speech Recognition with RNN based Acoustic Model," 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), 2019, pp. 563-566.
13) C. Shan, J. Zhang, Y. Wang and L. Xie, "Attention-Based End-to-End Speech Recognition on Voice Search," 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018, pp. 4764-4768.
14) J. Cui et al., "Improving Attention-Based End-to-End ASR Systems with Sequence-Based Loss Functions," 2018 IEEE Spoken Language Technology Workshop (SLT), 2018, pp. 353-360.
15) S. Sigtia, J. Bridle, H. Richards, P. Clark, E. Marchi and V. Garg, "Progressive Voice Trigger Detection: Accuracy vs Latency," ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 6843-6847.
16) H. Shim, D. Lowet, S. Luca and B. Vanrumste, "LETS: A Label-Efficient Training Scheme for Aspect-Based Sentiment Analysis by Using a Pre-Trained Language Model," in IEEE Access, vol. 9, pp. 115563-115578, 2021.
17) M. Aliramezani, E. Doostmohammadi, M. H. Bokaei and H. Sameti, "Persian Sentiment Analysis Without Training Data Using Cross-Lingual Word Embeddings," 2020 10th International Symposium on Telecommunications (IST), 2020, pp. 78-82.
18) R. MohammadiBaghmolaei and A. Ahmadi, "Word Embedding for Emotional Analysis: An Overview," 2020 28th Iranian Conference on Electrical Engineering (ICEE), 2020, pp. 1-5.
19) D. Goularas and S. Kamis, "Evaluation of Deep Learning Techniques in Sentiment Analysis from Twitter Data," 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications (Deep-ML), 2019, pp. 12-17.