Research Article

Shuyan Wang*

Recognition of English speech – using a deep learning algorithm

https://doi.org/10.1515/jisys-2022-0236
received September 25, 2022; accepted December 09, 2022

Abstract: The accurate recognition of speech is beneficial to the fields of machine translation and intelligent human–computer interaction. After briefly introducing speech recognition algorithms, this study proposed to recognize speech with a recurrent neural network (RNN) and adopted the connectionist temporal classification (CTC) algorithm to align input speech sequences and output text sequences forcibly. Simulation experiments compared the RNN-CTC algorithm with the Gaussian mixture model–hidden Markov model and convolutional neural network-CTC algorithms. The results demonstrated that the more training samples the speech recognition algorithm had, the higher the recognition accuracy of the trained algorithm was, but the training time consumption increased gradually; the more samples a trained speech recognition algorithm had to test, the lower the recognition accuracy and the longer the testing time. The proposed RNN-CTC speech recognition algorithm always had the highest accuracy and the lowest training and testing time among the three algorithms when the number of training and testing samples was the same.

Keywords: English, speech recognition, recurrent neural network, connectionist temporal classification

AMS Mathematics Subject Classification number: 68T07

1 Introduction

Speech is a form of communication in nature, and human beings have gradually created characters corresponding to their speech in the evolutionary process, and they have become important tools for human social communication [1]. Speech recognition technology aims to convert natural human speech into linguistic text that computers can understand [2]. Initially, speech recognition technologies were designed to recognize the pronunciation of individual words, but speech recognition of individual words alone can no longer meet the gradually increasing needs of human–computer interaction. The difficulty of speech recognition for sentences composed of plural words will increase due to the increased number of words to be recognized, the pronunciation habits of the speakers, and the environment [3]. Related studies on speech recognition techniques are as follows. Fantaye et al. [4] used a multilingual deep neural network (DNN) modeling approach for speech recognition. The experimental results showed that all multilingual models based on basic phonemes and rounded phoneme units outperformed the corresponding monolingual models. Prasad [5] compared two text classification algorithms for speech recognition, convolutional neural network (CNN) and recurrent neural network-long short-term memory (RNN-LSTM) algorithms, and found that the RNN-LSTM algorithm performed better in terms of discovery accuracy and precision than the CNN algorithm. Sun et al. [6] introduced an unsupervised deep domain adaptation acoustic modeling approach that jointly learned two discriminative classifiers with a DNN. The speech recognition experiments for noise/channel distortion and domain shift verified the effectiveness of the approach. To improve
the performance of neural machine translation in low-resource environments, Ahmadnia et al. [7] extended high-quality training data by generating pseudo-bilingual datasets and then using reverse translation-based quality estimation to filter low-quality alignments. Their study significantly improved machine translation performance. This study briefly introduced the traditional English speech recognition method and proposed recognizing English speech with an RNN algorithm. Finally, simulation experiments were conducted on the RNN-connectionist temporal classification (CTC)-based English speech recognition algorithm in MATLAB software, and it was compared with the other two English speech recognition algorithms, Gaussian mixture model–hidden Markov model (GMM–HMM) and convolutional neural network-CTC (CNN-CTC).

2 Recognition of English speech

2.1 Traditional English speech recognition method

The traditional English speech recognition process is shown in Figure 1. The first step is the acquisition of speech samples, followed by pre-processing of speech samples and feature extraction [8]. After the speech pre-processing, features are extracted from every frame of speech signal using the Mel-frequency cepstral coefficient method [9]. Then, an acoustic model converts the speech features into the most probable phoneme. The acoustic model is divided into two parts: one is to convert the speech features into the most probable state prediction value using the GMM [10], and the other is to convert the state prediction value into the most probable phoneme using the HMM. Finally, the phonemes are converted into words to form a sentence with the highest likelihood using a language model.

![Figure 1: Traditional English speech recognition process.](image)

2.2 English speech recognition based on deep learning

As mentioned above, the traditional English speech recognition method divides the recognition process into several stages, and the models used in different stages are independent of each other [11]. A complex decoder is required to decode the acoustic and language models in the process of actual use, which is tedious. In addition, the training of acoustic models in the traditional English speech recognition method also requires the forced alignment of features and states.

Compared with the traditional English speech algorithm, the deep learning-based English speech recognition algorithm integrates the acoustic and language models into a deep learning neural network [12], that is, integrates the parameters of the acoustic and language models into the parameters of the neural network. The parameters of the recognition model are optimized by training the neural network so that the speech features can be directly converted into words through the neural network. In the process of recognizing continuous speech, the audio signal of speech is first divided into frames before recognition; that is, the characters represented by every audio frame are recognized, and finally, the characters of every frame are combined together. However, even when the same person speaks the same sentence, the length of speech may vary; because of this, a single phoneme may be split into multiple frames, making the frame sequence
length of the audio much longer than the actual phoneme sequence; that is, the audio frame sequence cannot correspond to the actual phoneme sequence. Therefore, CTC is used to solve the problem of forced alignment between input and output sequences in traditional speech recognition algorithms [13]. The basic principle of the CTC algorithm for forced alignment is as follows. A blank character “-” is added to the candidate phonemes (characters) of the recognition model to make the recognition model get the candidate phoneme sequence frame by frame. The blank character is deleted after merging the consecutive repeated phonemes.

An RNN that can use historical information is used to recognize English speech [14]. Unlike a back-propagation neural network and a CNN, the hidden nodes in an RNN are computed by taking into account the input of the input layer at the current moment and the influence of the state of the hidden nodes at the last moment, in line with the temporal characteristics of the speech signal.

Figure 2 shows the training and use process of the deep learning-based English speech recognition algorithm. The detailed steps are as follows.

1. Like the traditional speech recognition algorithms, the input speech samples are processed by pre-emphasis, windowing [15], and feature extraction.
2. The input layer nodes of the RNN are input with the speech feature vector of one frame at one moment, and the number of input layer nodes depends on the dimension of the speech feature vector of one frame.
3. The calculation is performed in the hidden layer nodes of the RNN:
   \[
   \begin{align*}
   h_t &= f(w_{hh}h_{t-1} + w_{hh}x_t) \\
   y_t &= g(w_{hh}h_t),
   \end{align*}
   \]
   where \( h_t \) is the state vector of the nodes in the hidden layer at time \( t \), \( x_t \) is the speech feature vector input into the input layer at time \( t \), \( h_{t-1} \) is the state vector of the nodes in the hidden layer at the time \( t - 1 \), \( y_t \) is the output vector of the output layer at time \( t \) (the output vector of the whole output layer is the label probability distribution) [16], \( w_{hh} \) is the weight between the input and hidden nodes, \( w_{hh} \) is the weight between the hidden nodes, \( w_{hh} \) is the weight between the hidden and output nodes, \( f(\cdot) \) is the activation function of the hidden layer, and \( g(\cdot) \) is the activation function of the output layer.
4. After obtaining the label probability distribution in the output layer, it is determined whether it is currently in the training phase. If it is not in the training phase, the beam search algorithm [17] is used to decode the label distribution probabilities arranged in chronological order and output by the output layer to obtain the word sequence corresponding to speech.
5. If it is still in the training phase, the negative log-likelihood of the output label sequence is calculated in the CTC layer based on the corresponding label sequences in the training samples and the label
distribution probability sequences given in the output layer and used as the loss in the training process. The calculation formula is as follows:

$$L_{CTC} = -\ln \sum_u a_t^u \beta_t^u,$$

where $L_{CTC}$ is the training loss, $\alpha_t^u$ is the sum of the forward probabilities of label $u$ in the training sample corresponding to the label sequence at time $t$, and $\beta_t^u$ is the sum of the backward probabilities of label $u$ in the training sample corresponding to the label sequence at time $t$ [18].

⑥ It is determined whether the training is finished. If the training loss converges to stable or the number of iterations reaches the preset number, the training is finished; if the training is not finished, the weight parameters are reversely adjusted in the RNN according to the training loss, and then, it returns to step ③.

3 Simulation experiments

3.1 Experimental environment

MATLAB software [19] in the laboratory server was used to conduct simulation experiments on the deep learning-based English speech recognition algorithm.

3.2 Experimental setup

The speech dataset used to conduct the simulation experiments was the publicly available English speech recognition dataset TIMIT [20], which has sampling parameters of 16 kHz and 16 bits. There are 630 participants, and it includes 6,300 sentences. The phoneme level of every sentence was manually divided and labeled.

In the RNN-based English speech recognition algorithm, the relevant parameters of the RNN network are shown below. The number of nodes in the input layer was set as 39 according to the feature dimension obtained by the Mel-frequency cepstral coefficient method. The number of nodes in the output layer was set as 50 according to the number of phoneme tags and the blank and termination characters in the speech data set. The number of nodes in the hidden layer was set as 200. The activation function in the hidden layer was the sigmoid function, and the activation function in the output layer was set as the softmax function [21].

In order to further verify the recognition accuracy of the RNN-based English speech algorithm, this study also used GMM-HMM-based and CNN-based speech recognition algorithms for comparison. The parameters of the GMM-HMM-based speech recognition algorithm are shown below. The number of states of speech samples in the HMM model was set as 6, and the number of possible observations of every state was set as 4. The structural parameters of the CNN algorithm in the CNN-based speech recognition algorithm are shown below. The input specification of the input layer was set as $39 \times 1$. The number of nodes in the output layer was set as 50. The number of convolutional layers was set as 2, every convolutional layer had 32 convolutional kernels in a size of $2 \times 1$, and the sigmoid function was used in convolution calculation. The number of pooling layers was set as 1, the pooling box of every pooling layer was set as $2 \times 1$, the step length of the pooling box was 2 when sliding on the feature map, and mean-pooling was used in the pooling box.

The aforementioned algorithm parameters were obtained through orthogonal experiments.

3.3 Experimental items

⑦ The English speech data set was divided into a training set and a test set: 500, 1,000, 1,500, 2,000, and 2,500 sentences were set in the training set, and 2,000 sentences were set in the test set, respectively, to test the recognition performance of three speech recognition algorithms for English speech under different numbers of samples in the training set. The training and the test time were also recorded.
The English speech data set was divided into a training set and a test set; 2,500 sentences were set in the training set, and 500, 1,000, 1,500, 2,000, and 2,500 sentences were set in the test set. The English speech recognition performance of the three speech recognition algorithms was tested under different numbers of sentences in the test set. The training and the test time were also recorded.

### 3.4 Performance evaluation criteria

The word error rate [15] was used to evaluate the results after recognition by the speech recognition algorithm, and the calculation formula is as follows:

\[
\text{WER} = \frac{X + Y + Z}{P} \times 100\%,
\]

where \(X\) is the number of substituted words, \(Y\) is the number of missing words, \(Z\) is the number of inserted words, and \(P\) is the total number of words.

### 3.5 Experimental results

The recognition performance of the three speech recognition algorithms trained with different numbers of training samples after testing with the same number of samples in the test set is shown in Figure 3. Table 2 shows the training time of the three speech recognition algorithms under different numbers of samples in the training set and the test time. It was seen from Figure 3 that the recognition error rate of the three speech recognition algorithms under the same number of test samples gradually decreased as the number of training samples in the training set increased, while the recognition error rate of the GMM-HMM algorithm was the highest, the CNN-CTC algorithm was the second, and the RNN-CTC algorithm was the lowest when the number of training samples was the same.

![Figure 3: Recognition performance of three speech recognition algorithms trained on training sets with different numbers of samples.](image)

As could be seen from Table 1, the time spent by the three speech recognition algorithms in the training phase increased as the number of training samples in the training set increased; under the same number of training samples, the GMM-HMM algorithm took the most time to train, the CNN-CTC algorithm the second,
and the RNN-CTC algorithm the least, but the test time was not increased when they were tested by the test set. In addition, the GMM-HMM algorithm spent the longest time in testing, the CNN-CTC algorithm the second, and the RNN-CTC algorithm the least.

After training on a training set containing 2,000 training samples, the three speech recognition algorithms were tested for speech recognition on a test set with different number of samples. The variation of their word error rates with the number of test samples is shown in Figure 4. Table 2 shows the corresponding training and testing time. It was seen from Figure 4 that the word error rates of the three speech recognition algorithms increased as the number of test samples increased, and the word error rate of the GMM-HMM algorithm was the highest, the CNN-CTC algorithm was the second, and the RNN-CTC algorithm was the lowest under the same number of test samples.

![Figure 4: Recognition performance of three speech recognition algorithms for test sets with different numbers of samples.](image)

As could be seen from Table 2, since the number of samples in the training set was constant at 2,000 sentences, the training time of the algorithms before testing on the test set with different numbers of test
samples was nearly the same, and the training time of the GMM-HMM algorithm was the longest, followed by CNN-CTC and RNN-CTC algorithms. In terms of test time, the time consumed by all three speech recognition algorithms increased as the number of test samples increased; the test time consumed by the GMM-HMM algorithm was the longest, the CNN-CTC algorithm was the second, and the RNN-CTC algorithm was the least under the same number of test samples.

4 Discussion

Speech communication is a common form of human communication and one of the most convenient forms of communication. With the development of smart technology, various smart devices provide convenience for people’s daily life. Human–computer interaction between facilities and people is crucial to ensuring the convenience of smart facilities. Speech recognition technology is a kind of technology that can realize human–computer interaction. Smart devices cannot understand natural human language directly and should recognize speech before giving feedback to voice commands. In addition, speech recognition technology can also be applied to translating English speech. For these applications, the accuracy of speech recognition is very important. This study put forward to recognize English speech with the RNN and adopted the CTC algorithm to align input speech sequences and output text sequences forcibly. The RNN-CTC algorithm was compared with the GMM-HMM and CNN-CTC algorithms through simulation experiments. The final results have been shown above.

After training the three speech recognition algorithms using different sizes of training sets, the same test set was used to test the three algorithms. The results indicated that the larger the size of the training set, the higher the accuracy of the trained speech recognition algorithms, and the longer the training time; when the training set was the same, the RNN-CTC algorithm had the highest accuracy and the shortest testing time. The reason for these results was analyzed. With the increase in the number of training samples, the laws that the three speech recognition algorithms could fit became increasingly perfect, so their recognition accuracy for the test samples was higher, and the word error rate became lower. The RNN-CTC algorithm effectively utilized the temporal characteristics of speech, so it obtained the perfect recognition law after fitting. In terms of time consumption, with the increase in the number of training samples, the amount of data that needed to be processed by the recognition algorithms for training also increased; therefore, the training time consumption increased. The number of test samples did not change when they were tested by the test set, so the test time did not change either.

After training the three speech recognition algorithms with the training set of the same size, test sets with different sizes were used to test them. The results suggested that the larger the size of the test set, the lower the accuracy of the trained speech recognition algorithm, and the longer the testing time; when the number of test samples was the same, the RNN-CTC algorithm had the highest accuracy and the shortest test time. The reason for these results was analyzed. The increase in the number of test samples increased the recognition errors of the speech recognition algorithms, leading to an increase in the word error rate. As the recognition law fitted by the RNN-CTC algorithm utilized the temporal information of speech, the word error rate was the lowest. In terms of time consumption, because the number of training samples was constant, the training time of the algorithms was nearly the same before conducting the test; when the number of test samples increased, the recognition algorithm needed to process more data, so the test time increased accordingly.

5 Conclusion

This study briefly introduced the traditional English speech recognition method and proposed recognizing English speech with an RNN. Simulation experiments were performed on the RNN-CTC-based English
speech recognition algorithm in MATLAB software, and it was compared with the other two English speech recognition algorithms, GMM-HMM and CNN-CTC algorithms. The results are shown below. (1) With the increase in training samples, the recognition accuracy of all three English speech recognition algorithms increased; the word error rate of the GMM-HMM algorithm was the highest, the CNN-CTC algorithm was the second, and the RNN-CTC algorithm was the lowest under the same number of training samples. (2) As the number of test samples increased, the recognition accuracy of all three English speech recognition algorithms decreased, but the word error rate of the GMM-HMM algorithm was the highest, the CNN-CTC algorithm was the second, and the RNN-CTC algorithm was the lowest under the same number of test samples. (3) The increase in training and testing samples extended the training and testing time of the three speech recognition algorithms, but under the same number of training or testing samples, the GMM-HMM algorithm always spent the longest time, the CNN-CTC algorithm the second, and the RNN-CTC algorithm the shortest.

This study used the RNN to recognize phonemes of English speech and solved the problem of not being able to align one to the other due to the different lengths of speech sequences and text sequences by the CTC algorithm, which provides an effective reference for the improvement of English speech recognition technology. The shortcoming of this study is that only the RNN was tried to recognize English speech, but it is not ideal for long-sequence sentences, so the future research direction is to improve RNN.

Conflict of interest: The author declares no conflict of interest.

References


