Automated Speech Scoring with DNN or GMM in Goodness of Pronunciations

Yewei SONG

University of Birmingham, Edgbaston B15 2TT, UK
yxs987@student.bham.ac.uk

Abstract. This Mini-Project report investigates and analyzes speech scoring algorithms based on automatic speech recognition(ASR) technology. The reporter selected the deep neural network and the mixed Gaussian model from the development history of ASR technology as a comparison between the new and old technologies. Among various speech scoring algorithms, Goodness of prounciation was selected as the experimental algorithm. It also provides an analysis basis for these choices.

Keywords: Goodness of prounciation · Automatic speech recognition · Deep neural network

1 Introduction

Artificial intelligence is widely used in education. Using intelligent technology to score learning outcomes is a hot topic in education. Scoring can be mainly categorized into scoring of written content and scoring of speech, and all these rely on natural language processing technology. This article mainly discusses the speech scoring system based on ASR.
In order to score speech, we have to solve the following questions: How to convert speech into a sequence that the system can recognize? What kind of speech is "good"? How to model the scoring process? In order to solve the computer's speech recognition problem, we will use automatic speech recognition technology as the first step of the entire system. Then, according to the existing system design, we can find that there are two main scoring proposals. The first is to calculate the scores of various indicators of speech, and the second is to calculate the similarity between speech and native speakers (or speech models).

2 Previous Work

2.1 Automatic Speech Recognition

Speech recognition is a technology that converts input human speech into text or other meaningful label through a computer program. It is also known as automatic speech recognition (ASR), computer speech recognition or speech to text (STT).

In the early days of the development of computer technology, there was an idea to realize human-computer interaction directly by understanding human speech. Automatic speech recognition, as a classic subject in computer science, has a high demand for commercial applications. But human language itself is rich in change, highly complex and ambiguous. In the 20th century, the major speech recognition systems supported few languages, poor adaptability to dialects, slow recognition speed, and low accuracy. At the same time, people have high expectations for speech recognition systems, as in the films and televisions that command computers and robots through speech.

Speech recognition integrates the knowledge of many different disciplines, spanning basic and cutting-edge disciplines such as mathematics and statistics, acoustics and linguistics, computer science and artificial intelligence, and is also related to psychology and cognitive science. [4]

The first automatic speech recognition system that successfully identified the digits 0 to 9 in 1952 by Davis and his colleague [9]. Before 2009, speech recognition was mainly based on Gaussian Mixture Model Hidden Markov Model (GMM-HMM) machine learning technology, and the accuracy of speech recognition improved slowly. The Deep Neural Network Hidden Markov Models (DNN-HMM) technology that has been widely used in the past decade has greatly improved the accuracy of speech recognition. In 2017, Microsoft achieved a word error rate of only 5.1% on Swichboard [1].

First of all, we know that the audio collected through the microphone is a digital wave. The most common audio formats include .wav uncompressed waves and .mp3 compressed audio. Before any speech recognition process, we need to convert the audio file into a digital wave format like Figure 1.

Fig. 1. Wave File

Then, we need to cut off the mute at the beginning and end of the audio to reduce the interference to the next steps. Then we need to "frame" the sound, it means cut the sound into small pieces, each piece is called a frame, usually one piece include 10-30ms voice. Frame audio is not cut into a simple time slice, but usually uses a windowed Fourier transform called Window function.

The small piece of audio obtained after framing lacks the ability to describe information in the time domain, meaning it is difficult to identify as a special feature. So we need to transform the waveform into features that can be processed. The most mentioned transformation method is called Mel-scale Frequency Cepstral

Coefficients (MFCC) [21] [32], but there are more effective ways like beamformering DNN etc. The principle of this method is to transform each frame waveform into a multi-dimensional vector according to the physiological characteristics of the human ear. This entire process is called acoustic feature extraction. [16]
Before we convert this matrix of N rows (N represents the number of extracted dimensions) and M columns (M represents the length of time) into text, we need to introduce two concepts.

* **Phonemes** The pronunciation of a word consists of phonemes. A commonly used phoneme set is a set of 39 phonemes made by Carnegie Mellon University. [18]
* **States** A phoneme usually includes three states (left, middle, right). One states might be recognised from a few frames. We have designed some algorithms to guess what phoneme is stats belong to.

According to the above concepts, speech recognition can be summarized into three steps. The first step converts frames into states, the second step combines states into phonemes, and the third step combines phonemes into words. [27]

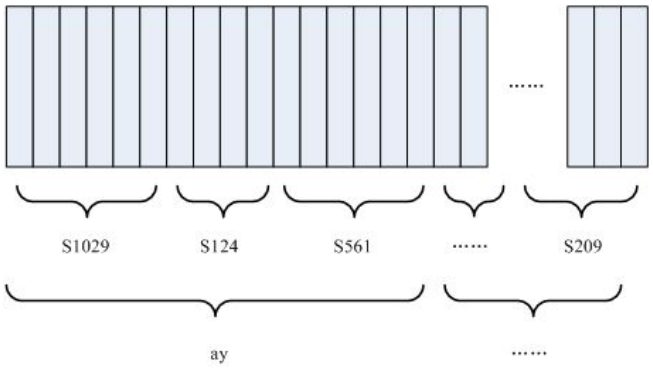

Fig. 2. Frame to Stats to Phonemes

In the Figure 2, each vertical bar represents a frame, several frames of speech represent a state, three states represent a phoneme, and several phonemes represent a word. If we know the states of each frame, we can infer the phonemes of the voice.
In order to identify the corresponding state and phoneme of each frame, we have previously trained and stored some parameters for predicting the probability. These parameters are called acoustic models. [27] When we have the acoustic features and the predicted probabilities of the corresponding phonemes, the next question is how to identify a sequence as another sequence of different length, such as correctly identifying an MFCC feature sequence as a phoneme. To solve

the problem we can use Hidden Markov Models.

2.2 HMM-GMM Model

GMM in phonemes recognition Gaussian Mixture Models (GMM), which is simply a superposition of multiple Gaussian distributions.

$$p(x) = \sum_{m=1}^{M} c_m N(x; \mu_m, \sigma_m^2) \tag{1}$$

Each feature is determined by a phoneme, that is, different features can be clustered by phoneme. In actual speech, a phoneme is related to the phonemes before and after it, so GMM will output a triphone composed of three phonemes. Sometimes phones are considered in context. There are triphones or even quinphones. But note that unlike phones and diphones, they are matched with the same range in waveform as just phones. They just differ by name. That's why we prefer to call this object senone. A senone's dependence on context could be more complex than just left and right context [15].
Phonemes are represented as hidden variables (states) in HMM, they are equivalent to:
Each feature is determined by several states, that is, different features can be clustered by state.
GMM mixes states as the probability distribution of different phonemes [28]. The trained GMM can calculate the hidden variables of stats. We can also understand GMM as a simple classifier that outputs the probability combination of the states as different phonemes. At the same time, we can replace it with other classifiers suitable for time sequence data, such as deep neural networks.

HMM in acoustic model In speech recognition, we assume that each word or phoneme corresponds to a hidden Markov model. Hidden Markov Model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobservable states. Suppose f(t) is a time series, and Markov Chain is a stochastic process that assumes that f(t + 1) is only related to f(t). The difference between HMM and simple Markov model is that state is not directly visible, but some variables affected by state are visible. Each state has a probability distribution over the possible outputs. Therefore, the output sequence can reveal some information about the state sequence. [3]
In the decoding problem, given the number of hidden states, transition probability, and visible state chains, solve the hidden state chains. In actual calculations, we do not calculate the probability of each possible state sequence. The Viterbi route-finding algorithm is usually used.
To calculate the Viterbi route [13], suppose the state space of a given hidden Markov model (HMM) is S, There are k states, and the probability of the initial state i is π_i, The transition probability from state i to state j is $a_{i,j}$, Say

we observe outputs $y_1, \cdots, y_T$. The most likely state sequence that $x_1, \cdots, x_T$ produces the observations is given by the recurrence relations:

$$V_{1,k} = P(y_1|k) \cdot \pi_k \tag{2}$$
$$V_{t,k} = max_{x \in S}(P(y_1|k) \cdot a_{x,k} \cdots V_{t-1,x}) \tag{3}$$

$V_{t,k}$ is the probability of the most probable state sequence $P(x_1, \ldots, x_t, y_1, \ldots, y_t)$ responsible for the first t observations that have k as its final state. The Viterbi path can be retrieved by saving back pointers that remember which state x was used in the second equation. Let $Ptr(k, t)$ be the function that returns the value of x used to compute $V_{t,k}$ if $t > 1$, or k if $t = 1$. Then:

$$x_T = argmax_{x \in S}(V_{T,x}) \tag{4}$$
$$x_{t-1} = Ptr(x_t, t) \tag{5}$$

Here we're using the standard definition of arg max. [19] [31]

Forced Alignment After calculating a Viterbi route in HMM, we get a sequence that might include many consecutive and identical phonemes. The obvious problem is different speech has different speed, a hello might be 'hhhhhheeeeellllllloooo' or 'heeeeellooooo', but they are same word and we call it phoneme loop. Describe into time sequence model, the number of inputs is more than the number of outputs. How is the input and output aligned?
To solve this problem, the most common solution is **Forced Alignment**. The core problem of forced alignment is how to convert different sequences containing different numbers of consecutive repeating elements into the same sequence. In 2006, Alex Graves proposed the Connectionist Temporal Classification, an improved neural network that can calculate the probability distribution of all possible alignments [14]. Suppose the input sequence is $X = [x_1, x_2, \cdots, x_T]$ and output sequence is $Y = [y_1, y_2, \cdots, y_U]$, the alignment problem is mapping X to Y. After train the neural network, we can get a predict that $Y^* = \arg\max_Y p(Y|X)$. In result the example phoneme sequence will all be predict as 'hello'. The whole workflow in ASR can be summary in Figure 3

2.3 Deep-Learning and ASR

What is Deep learning? Machine learning is a discipline that specializes in how computers simulate or implement human learning behaviors in order to acquire new knowledge or skills and reorganize existing knowledge structures to continuously improve their performance. [24] Although machine learning has developed for decades, there are still many problems that are not well solved, such as image recognition, speech recognition, natural language understanding, weather prediction, content recommendation, etc. Common machine learning

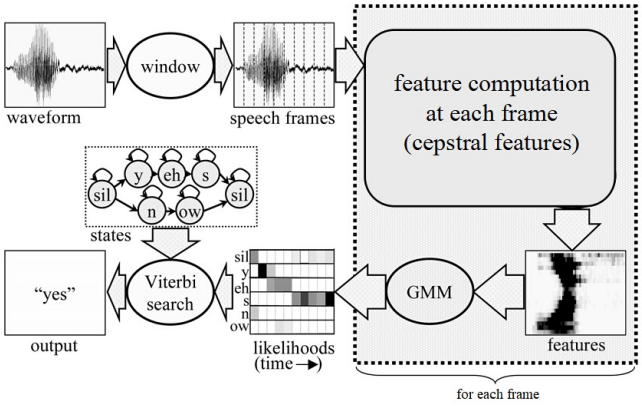

Fig. 3. Speech recognition, a big framework [5]

workflows include data collection, preprocessing, feature extraction, feature selection, and then inference, prediction, or recognition. But only the last half uses machine learning techniques. The middle part, also called feature expression, is very dependent on manual implementation.

The term Deep Learning was introduced to the machine learning community by Rina Dechter in 1988 [10], and to artificial neural networks by Igor Aizenberg and colleagues in 2000, in the context of Boolean threshold neurons. Deep Learning is based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. The foundation of deep learning is distributed representation in machine learning. Distributed representation assumes that observations are generated by the interaction of different factors. On this basis, deep learning further assumes that this interactive process can be divided into multiple levels, representing multiple levels of abstraction of observations. Different layers and scales can be used for different levels of abstraction. They also improved neural network acoustic models use Rectifier Nonlinearities. [23]

Deep learning uses this hierarchical abstraction, and higher-level concepts are learned from lower-level concepts. This layered structure is often built layer by layer using greedy algorithms and picking out more effective features that help machine learning. [12]

What problems can use Deep learning? Compared with the classic Machine Learning technology, Deep Learning provides a more powerful prediction model, which can usually produce good prediction results. Especially for data with complex features such as images and speech, with spatial and temporal continuity. Experiments on common data sets for testing, such as TIMIT in speech recognition and ImageNet in image recognition, Cifar10, prove that deep

learning can improve the accuracy of recognition. [11] [20]

Introduction of DNN-HMM According to the above, we can know that GMM-HMM has been the mainstream speech recognition technology. Using GMM classification features has advantages, GMM training is fast, acoustic models are small, and it is easy to transplant to embedded platforms. But, GMM does not use the context information of the frame, nor can it learn deep nonlinear feature transformations. [29]
In 2013, L Deng (Microsoft) used a Feed Forward Deep Neural Network (FFDNN) replacing GMM to build acoustic models [11]. After that, the old-style GMM-HMM is outdated. With the improvement of the general computing power of GPUs, more and more studies have begun to use deep learning to predict and classify frames. Many researchers have used a variety of network structures such as FFDNN, Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), Long Short Term Memory (LSTM) networks to model the output probability, resulting in many DNN-HMM acoustic models such as FFDNN-HMM, CNN-HMM, RNN-HMM, LSTM-HMM and have achieved effective results.
In the FFDNN-HMM modeling framework, the input features use the method of stitching frames around the current frame to build a model of the time series signal, and the model output maintains the triphone senone that is often used by GMM-HMM [8] and explained at section 2.2. Although FFDNN-HMM models the context information by framing, after all, the number of stitching frames is limited and the modeling ability is not strong. Therefore, the introduction of RNN enhances the long-term modeling ability. In addition to receiving the output of the previous hidden layer, the input of the hidden layer also receives the output of the previous hidden layer as the current input. [22] Through the cyclic feedback of the hidden layer of the RNN, the long-term historical information is retained and the model is greatly enhanced. The ability of memory and timing of speech is also well described by RNN. Speech recognition is based on the speech spectrum after time-frequency analysis, and the time-frequency spectrum of speech has structural characteristics. In order to improve the speech recognition rate, it is necessary to overcome the diversity of speech signals, including the diversity of speakers (speakers themselves and between speakers), and the diversity of environments. CNN provide translation-invariant convolutions in time and space. Applying the idea of convolutional neural networks to the acoustic modeling of speech recognition, you can use the invariance of convolution to overcome the diversity of speech signals themselves. [2] In simple terms, it can be considered that the time-frequency spectrum obtained by analyzing the entire speech signal is treated as an image, and a deep convolution network widely used in the image is used to identify it.

2.4 Speech Scoring

Unrestricted Spontaneous Speech A lexical matching method (Vector Space Model) and two semantic similarity measures (Latent Semantic Analysis and Pointwise Mutual Information) are used by Xie et al [37]. First they proposed three main scoring features:

- Sim_{max}: the score point which has the highest similarity score between test response and score vector.
- Sim_4: the similarity score to the responses with the highest score category.
- Sim_{cmb}: the linear combination of the similarity scores to each score category.

This method effectively calculates the similarity between the input speech and the standard speech, but according to the experimental results of this document, the accuracy of the annotation has a huge impact on speech recognition. The difference is particularly reflected in the ASR generated text and manual transcript. This means that we need more data preparation, especially manual transcripts, which do not meet the purpose of designing an automated scoring system to reduce manual workload.
But we can still understand the main dimensions and means of speech comparison from these similarity comparison algorithms.

Classify Algorithm The most intuitive way to solve the problem of speech scoring is to directly use the quantitative indicators to score the speech.In 2006, Klaus Zechner and Isaac I. Bejar summarized some major quantifiable metrics for speech scoring [38]. The main variables compared include:

- Lexical counts and length measures
- Lexical sophistication (unique words)
- Fluency measures (based on pause information)
- Rate measures (like word or types per second)
- Content measures (word vectors)

These dimensions include many of the main dimensions for evaluating speech, and are easy to calculate in the ASR system. They used SVM, multiple regression and classification trees to classify audios. However, because there is not enough linkage between indicators, candidate can easily use special tricks to deceive the system. A typical example is using a faster but inaccurate speaking rate to get a higher fluency score, because the system calculates fluency mainly based on the time and frequency of silence. Another common problem is that we need the pronunciation of native speakers of the test content to train a suitable model.

SpeechRater In 2008, the Educational Testing Service (ETS) hosting Test of English as a Foreign Language (TOEFL) designed a new system called SpeechRater, which is used by prospective test takers to prepare for the official TOEFL iBT

test [7]. They selected 29 features as the dimensions of judgment and divided them into three types as Delivery, Language Use and Topic Development described in Figure 4.

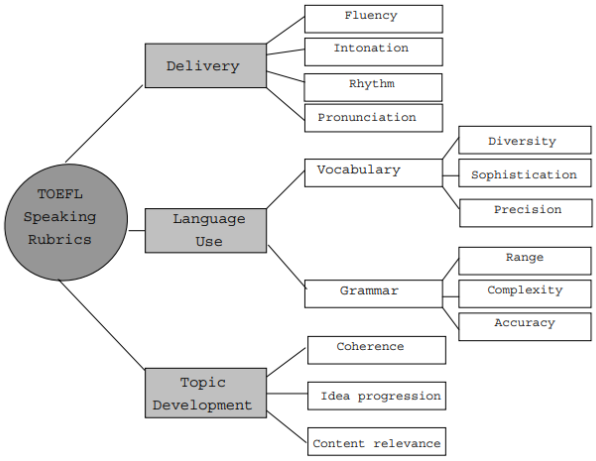

Fig. 4. Features that SpeechRater used

Based on future selection, in particular, the construct representation of the multiple regression model with expert weights was sufficiently broad to justify its use in a low-stakes application [36]. But the high-level speech components such as content and organization are ignored.

Goodness of Pronunciations Goodness of Pronunciation proposed by Witt [35] is a scoring algorithm with the core theory based on maximum likelihood. That is

$$\begin{aligned} GOP_1 &\equiv \left|log\left(P\left(p|O^p\right)\right)\right| / NF(p) \\ &= \left|log\left(\frac{P(O^p|p)P(p)}{\sum_{q\in Q} P(O^p|q)P(q)}\right)\right| \Big/ NF(p) \end{aligned}$$

where Q is the set of all phone models and NF(p) the number of frames in the acoustic segment O(p). The GOP algorithm mainly calculates the difference between the HMM model of the original text and the user's pronunciation, and it measures "the probability that this speech corresponds to the phoneme Q_i when the user's speech O is observed. [34]
In many scenarios, GOP performs well, but it is very dependent on the quality of the model. It seems that correct speech with a non-learned accent is treated the same way as speech with the accent of a non-native. So in order to improve the

adaptability of the system to different accents, we need to expand the amount of training data as much as possible.

Choices Goodness of Pronunciation performs well in most cases, and the algorithm itself pays more attention to the differences between pronunciation and ASR models. In other methods, if the algorithm requires a certain class of labeled training data, we cannot complete the model training and adaptation in a limited project time. Other feature recognition-based scoring algorithms are not in line with our project purpose, a non-deceivable and versatile speech scoring algorithm. Applicants might defraud score by increasing the speed of speech, imitating continuous reading, and ignoring grammatical errors while using system based on those algorithms. [30]

2.5 Critical Analysis

By studying previous speech scoring algorithms, we found that we can divide the algorithms into the following categories:

- **Feature-based selection** Comprehensive evaluation algorithm based on audio characteristics and transliteration results. The algorithm can combine the linguistic characteristics of speech content and the characteristics of speech to comprehensively score, but a single aspect is easily deceived.
- **Classification based on machine learning** The results of speech feature extraction are trained by a deep learning classifier, and then the training model is used to score new speech. However, the training process of the algorithm requires a large number of scored voices. Only ETS or the British Council have sufficient data.
- **Model-based comparison** Comparing and scoring the HMM model extracted from speech and the HMM model of the original content depends on two existing resources: the transliterated text of the speech and the acoustic model

For all feature-based technologies, we can first ask a question. Is speech with a good indicator of some features good speech? Obviously not. Consider a typical counterexample: if a person only has a good speech speed, rich vocabulary, and a long single sentence length, but has no grammar and coherent meaning, such speech is not good. These indicators are indeed important speech level scoring standards, but the overall evaluation should consider the worst parts.
Classification training or comparison of acoustic models of speech is another main idea, which is also the most intuitive answer to the scoring problem. Deep learning technology improves the ability to extract acoustic features, and a series of machine learning methods that solve time series data also solve some difficulties in the ASR field. But we can also ask the question whether the black box in the middle is "fair" from the acoustic characteristics to the evaluation of the speech level.
When constructing the model, we also need to discuss the following issues. The

first issue is the running cost. If a system needs to consume too many computing resources for scoring, and even cannot achieve real-time scoring, then we can consider that the cost of this system is too high. The second issue is immunity. There are always various kinds of interference in the system, such as lossy compression, noise, volume differences, and poor hardware contact. How to improve the robustness of the system is also a problem.

3 Experiment

3.1 Experiment Design

Based on the model selection above, we designed a simple implementation and verification of the Goodness of Pronunciation algorithm. The basic environment of the experiment is based on macOS Mojave operating system, Kaldi-ASR 5.5.636 with nnet3 and the Librispeech database (960 hours data). Then, in order to compare the different feature extraction methods, especially the impact of GMM and DNN on the speech score, scores using two different acoustic models were compared.

3.2 Data Preparation

Assumed that the IELTS speaking test can effectively grade the candidate's speech level [17]. Design a data collection object with 3 gradients, about 3 people in each group: Level A (Beginner) IELTS Speaking not more than 5.5; Level B (Competent) between 6.0 and 6.5; Level C (Skilled user) 7 and higher. Convert the recorded test audio into .wav format with a uniform sampling rate (44100Hz). To train the acoustic model, use nnet3 as basic neural networks; use LibriSpeech as the default training data; pre-train two comparison models; use full 960 hour data [25].

3.3 Model Train and Test Procedure

The entire experimental execution process can be divided into three parts. The first part is training the acoustic model. The first step was to download 960 hours of labeled voice files from Librispeech. The second step is to use Kaldi's built-in training script on librispeech which is in the column *egs/librispeech/s5/run.sh*. You must fix the data parameter to your own data location. The third step is to export the trained model for subsequent experiments. It includes .mdl, .fst and word dictionary file et al. The GMM model of librispeech is offered online at `kaldi-asr.org/downloads/build/6/` . Also you can also use scripts for training, the order is *gmm-init-mono, compile-train-graph, align-equal-compiled, gmm-acc-stats-ali* and *gmm-est*.
The second part is preprocessing of test data. The first step is some processing for the audio itself, including removing noise, removing white space, and adjusting the volume. Then prepared the test data according to Kaldi's Data

preparation document [26].
The third part is the test execution process, and two speech scoring processes will be performed according to the purpose of the experiment. The feature extraction using both DNN (nnet3), and GMM. The DNN feature extraction is used *nnet3-compute* function and GMM feature extraction is used *compute-mfcc-feats* function. Next step is use the *steps/nnet/align.sh* script to generate the output of the forced-alignment of the learner's uttered speech. All above scripts and program are include in the Kaldi and documents in the `kaldi-asr.org/doc`. The output file is a .ark file of frame level posterior-probabilities and a .txt file of alignment. Finally, we have the the score is obtained according to the GOP algorithm which implemented in Appendix A. The workflow of the entire experiment is shown in Figure 5.

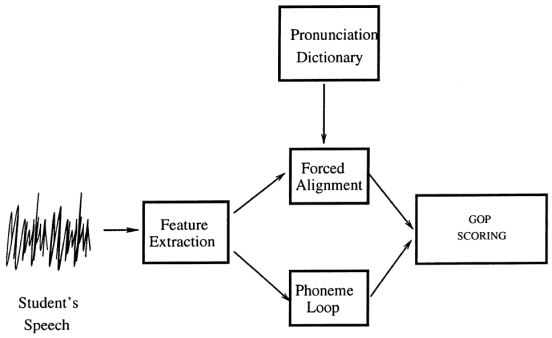

Fig. 5. The workflow of the experiment

3.4 Test Results

Due to the short duration of the project, only 6 groups of samples were collected in this experiment from 6 non-native English learners. GoP values and the calculated average are given in Table 1.
Comparing the differences in GOP scores between different CEFR levels, we

Table 1. Average GOP score per CEFR level

CEFR Level	GMM-HMM GOP	DNN-HMM GOP
C1+	0.814	0.809
B2	0.610	0.691
B1	0.607	0.595

can see that higher CEFR levels have significantly better pronunciation levels

in the selected samples. At the same time, the DNN model seems to show better discrimination in the speech scores of the middle and low scores. And DNN model has larger score differences between learners at different levels
In addition, throughout the experiment, the designer found several interesting facts. The first, Non-native English speakers at a medium level differ greatly in their speaking ability. One tester achieved an IELTS score of 6.5, but the system score was lower than another 6-point learner. Of course, this may also be due to the difference in the scores of different test centers and examiners, which can be improved by obtaining a larger amount of data. Second, due to the lack of Asian accents in the training data, many syllables cannot be effectively identified. This can be used as a counter-example to illustrate the dependence of the GOP algorithm on the model.

4 Conclusion

4.1 Experiment Conclusion

Based on the results of the experiment, we can think that GOP can effectively score speech, and the gradient of the score is more obvious between high-level language learners and intermediate-level learners.DNN also has better performance than GMM in the test of experimental data, it extracts more correct phonemes.

4.2 Investigate Conclusion

Based on research on literature reading and technical implementation, the author summarizes the following points:
First, the use of DNN instead of GMM for feature extraction has been widely accepted. With the improvement of hardware support for neural networks, the cost of using neural networks in real-time systems has gradually decreased.
Second, neural networks can be used not only for feature extraction, but also for forced alignment, scoring systems, etc. The methods and theories of applying DNN in these speech recognition links are still not perfect, which is a promising research direction.

4.3 Expectations

In this Mini-Project, the following questions arise and author proposed some expectations to solve them:
Insufficient data Due to the short project period, it is difficult to collect and preprocess many voices. In the SpeechRater product development of ETS, they used about 100 hours of the candidate's voice to conduct the experiment. Assuming that each candidate provides 10 minutes of non-blank speech, about 600 pieces of experimental data are acceptable [36]. And also the voice should covering learners at all levels from IELTS 4.0 to 8.0, and using native speakers voices

as the control group.
Lack of model diversity The first reason is that there are not enough computers for model training, and the second reason is the lack of data for the Asian accent training set. If we can verify the impact of models on scoring, there is reason to persuade scoring agencies to collect Asian accents for training models.
Reliability of speech classification IELTS is a general annotated English proficiency test, but the score of its oral test is subjectively influenced by the examiner. If double-blind experimental data based on the hearing of native speakers can be used as label, subjective effects can be avoided to the greatest extent.

4.4 Future Works

From this Mini-Project research process, I explored the following related research directions, these directions are suitable for summer projects.

Question and answer system in the field of education An article published by Boe in 2006 pointed out that after 2000, the shortage of certified teachers has greatly increased [6]. At the same time, in Australia, kindergarten teachers are also one of the scarce occupations [33]. Using artificial intelligence to assist teachers in teaching is an effective way to alleviate the shortage of teachers. In order to simulate teaching interaction, the question answering system is an important part of computer-aided teaching. This includes questions raised by students, as well as questions raised by computers. The former is for answering doubts, the latter is for testing. This research can explore a question-and-answer system that can be applied to assist teaching, and can evaluate the learning effect of students.

Automatic correction of text assignments In the study of Mini-Project, we explored the pronunciation scoring system. But we found that how to evaluate the high-level speech components is difficult. Here the author hopes to be able to use the ASR transcribing results as text assignments with some spelling errors and explore suitable NLP methods to evaluate them. Applying the results of the text evaluation to the total evaluation results can improve the accuracy of the speech score.

References

1. The microsoft 2017 conversational speech recognition system. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5934–5938. IEEE (2018)
2. Abdel-Hamid, O., Mohamed, A.R., Jiang, H., Penn, G.: Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition. In: 2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4277–4280. IEEE (2012)

3. Bahl, L., Brown, P., De Souza, P., Mercer, R.: Maximum mutual information estimation of hidden markov model parameters for speech recognition. In: ICASSP'86. IEEE International Conference on Acoustics, Speech, and Signal Processing. vol. 11, pp. 49–52. IEEE (1986)
4. Benzeghiba, M., De Mori, R., Deroo, O., Dupont, S., Erbes, T., Jouvet, D., Fissore, L., Laface, P., Mertins, A., Ris, C., et al.: Automatic speech recognition and speech variability: A review. Speech Communication **49**(10-11), 763–786 (2007)
5. Bidgoli, H.: Encyclopedia of Information Systems. Academic Press (2002)
6. Boe, E.E., Cook, L.H.: The chronic and increasing shortage of fully certified teachers in special and general education. Exceptional Children **72**(4), 443–460 (2006)
7. Chen, L., Zechner, K., Yoon, S.Y., Evanini, K., Wang, X., Loukina, A., Tao, J., Davis, L., Lee, C.M., Ma, M., et al.: Automated scoring of nonnative speech using the SpeechRater sm v.5.0 engine. ETS Research Report Series **2018**(1), 1–31 (2018)
8. Dahl, G.E., Yu, D., Deng, L., Acero, A.: Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. IEEE Transactions on Audio, Speech, and Language Processing **20**(1), 30–42 (2011)
9. Davis, K.H., Biddulph, R., Balashek, S.: Automatic recognition of spoken digits. The Journal of the Acoustical Society of America **24**(6), 637–642 (1952)
10. Dechter, R., Pearl, J.: Network-based heuristics for constraint-satisfaction problems. In: Search in Artificial Intelligence, pp. 370–425. Springer (1988)
11. Deng, L., Hinton, G., Kingsbury, B.: New types of deep neural network learning for speech recognition and related applications: An overview. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. pp. 8599–8603. IEEE (2013)
12. Deng, L., Yu, D., et al.: Deep learning: Methods and applications. Foundations and Trends® in Signal Processing **7**(3–4), 197–387 (2014)
13. Forney, G.D.: The Viterbi algorithm. Proceedings of the IEEE **61**(3), 268–278 (1973)
14. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd International Conference on Machine Learning. pp. 369–376 (2006)
15. group, C.: Basic concepts of speech recognition. `https://cmusphinx.github.io/wiki/tutorialconcepts/`
16. Hermansky, H., Ellis, D.P., Sharma, S.: Tandem connectionist feature extraction for conventional hmm systems. In: 2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100). vol. 3, pp. 1635–1638. IEEE (2000)
17. Iwashita, N., Brown, A., McNamara, T., O'Hagan, S.: Assessed levels of second language speaking proficiency: How distinct? Applied Linguistics **29**(1), 24–49 (2008)
18. Lenzo, K.: The CMU pronouncing dictionary. `http://www.speech.cs.cmu.edu/cgi-bin/cmudict`, accessed April 4, 2010
19. Levinson, S.E.: Continuously variable duration hidden markov models for automatic speech recognition. Computer Speech & Language **1**(1), 29–45 (1986)
20. Li, L., Zhao, Y., Jiang, D., Zhang, Y., Wang, F., Gonzalez, I., Valentin, E., Sahli, H.: Hybrid deep neural network–hidden markov model (dnn-hmm) based speech emotion recognition. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction. pp. 312–317. IEEE (2013)
21. Logan, B., et al.: Mel frequency cepstral coefficients for music modeling. In: Ismir. vol. 270, pp. 1–11 (2000)

22. Lu, L., Zhang, X., Renais, S.: On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition. In: 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5060–5064. IEEE (2016)
23. Maas, A.L., Hannun, A.Y., Ng, A.Y.: Rectifier nonlinearities improve neural network acoustic models. In: Proc. icml. vol. 30, p. 3 (2013)
24. Michie, D., Spiegelhalter, D.J., Taylor, C., et al.: Machine learning. Neural and Statistical Classification **13**(1994), 1–298 (1994)
25. Panayotov, V., Chen, G., Povey, D., Khudanpur, S.: Librispeech: an asr corpus based on public domain audio books. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5206–5210. IEEE (2015)
26. Povey, D.: Kaldi data preparation. `https://kaldi-asr.org/doc/data_prep.html`
27. Rabiner, L.R.: A tutorial on hidden markov models and selected applications in speech recognition. Proceedings of the IEEE **77**(2), 257–286 (1989)
28. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Processing **10**(1-3), 19–41 (2000)
29. Shahin, M., Ahmed, B., McKechnie, J., Ballard, K., Gutierrez-Osuna, R.: A comparison of GMM-HMM and DNN-HMM based pronunciation verification techniques for use in the assessment of childhood apraxia of speech. In: Fifteenth Annual Conference of the International Speech Communication Association (2014)
30. Strik, H., Truong, K., De Wet, F., Cucchiarini, C.: Comparing different approaches for automatic pronunciation error detection. Speech Communication **51**(10), 845–852 (2009)
31. Varga, A.P., Moore, R.K.: Hidden Markov Model Decomposition Of Speech And Noise. Baseline pp. 845–848 (1990)
32. Vergin, R., O'Shaughnessy, D., Farhat, A.: Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. IEEE Transactions on Speech and Audio Processing **7**(5), 525–532 (1999)
33. Warrilow, P., Fisher, K.: Early childhood teachers and qualified staff shortage. In: 8th Australian Institute of Family Studies Conference. Melbourne, Australia (2003)
34. Witt, S.M., Young, S.J.: Phone-level pronunciation scoring and assessment for interactive language learning. Speech Communication **30**(2-3), 95–108 (2000)
35. Witt, S.M., et al.: Use of speech recognition in computer-assisted language learning. Ph.D. thesis, University of Cambridge Cambridge, United Kingdom (1999)
36. Xi, X., Higgins, D., Zechner, K., Williamson, D.M.: Automated scoring of spontaneous speech using speechratersm v1. 0. ETS Research Report Series **2008**(2), i–102 (2008)
37. Xie, S., Evanini, K., Zechner, K.: Exploring content features for automated speech scoring. In: Proceedings of the 2012 conference of the North American chapter of the Association for Computational Linguistics: Human language technologies. pp. 103–111. Association for Computational Linguistics (2012)
38. Zechner, K., Bejar, I.I.: Towards automatic scoring of non-native spontaneous speech. In: Proceedings of the Main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics. pp. 216–223. Association for Computational Linguistics (2006)

A Part of Experiment Code (GoP)

```
for x in range(len(number_of_segments)):

    req_t_id=transition_id[cum_number_of_segments[x]:(
        cum_number_of_segments[x+1])];
    req_t_id.sort();
    score=0.0;

    for y in range(len(req_t_id)-1):

        tmp_prob = float(lookup_tab[req_t_id[y]-1][2]);
        tmp_pdf = int(lookup_tab[req_t_id[y]-1][1]);
        tmp_post = float(posterior[cum_number_of_segments
            [x]+y][tmp_pdf]);
        score = score + math.log(tmp_prob) + math.log(
            tmp_post);

    tmp_pdf = int(lookup_tab[req_t_id[-1]-1][1]);
    tmp_post = float(posterior[cum_number_of_segments[x
        +1]-1][tmp_pdf]);
    score = (score + math.log(tmp_post) + float(
        number_of_segments[x]-1)*math.log(num_of_senones))
         / float(number_of_segments[x]);
    phone_score.append(score);
```