International Journal of Web Applications

Print ISSN: 0974-7710
Online ISSN: 0974-7729

DLINE JOURNALS

IJWA 2024; 16 (2)
https://doi.org/10.6025/ijwa/2024/16/2/44-51

Construction of Viterbi Algorithm Voice Command Model for Autonomous Learning of College English

Woyou Zhang
Macau University of Science and Technology
Wei Long Road, Taipa, Macau
China
Ggfert545435@protonmail.com

ABSTRACT

This article presents a novel approach to college English autonomous learning-a voice command model based on the Viterbi algorithm. This innovative model is designed to enhance students' English proficiency and learning outcomes in an autonomous learning environment. The article begins by outlining the context and importance of the model's development, then delves into the key technologies utilized in its construction, such as speech recognition, the Viterbi algorithm, and feature extraction. The article concludes with a validation of the model's effectiveness and superiority through experiments, demonstrating its ability to accurately interpret students' voice commands and thereby improve the efficiency of college English autonomous learning.

Received: 2 November 2023
Revised: 9 January 2024
Accepted: 17 January 2024
Copyright: with Author(s)

Keywords: ESP Theory, College English, Autonomous Learning

1. Introduction

After two years of English study, the undergraduates' language ability did not perform well in practice. Therefore, it is urgent and necessary to explore the new college English teaching system, course setting and course contents and methods [1]. Thus, this article put forward some tentative ideas on the setting of the curriculum system of College English, especially how to consolidate the English knowledge gained by students in the third and fourth grades to improve their English ability further [2]. ESP teaching objectives are clear; teaching should focus on the purpose of learners. ESP teaching is based on the analysis of learner needs. The selection of teaching content and the adoption of teaching methods should be decided by students' learning needs [3]. ESP Teaching pays attention to the cultivation of pragmatic competence. Students master English mainly in their areas of expertise and skilled use to achieve their English communication needs. This theory has important guiding significance for college English teaching [4]. After the students pass CET-4, it is necessary to focus on individualised training of students and

set up autonomous learning centres in multimedia language laboratories as much as possible to achieve man-machine dialogue and improve students' enthusiasm for learning and learning efficiency [5]. Correspondingly, to reform college English teaching and meet the needs of the individualized development of students, we need to establish a sound follow-up English course system to meet the needs of students in different levels of language courses and further meet the needs of individualized development of students.

2. State of the Art

"English for Specific Purposes" is a new discipline developed in the 1960s whose educational aims are to meet the needs of different learners. Due to its clear objectives, strong pertinence, and high practical value, it is greatly welcomed by all kinds of English learners at all levels and the professionals and linguists of ESP. Foreign research on ESP has already been fruitful, and ESP is increasingly popular in China. Speech recognition technology, also known as automatic speech recognition, aims to convert the vocabulary of human speech into computer-readable input such as keystrokes, binary encodings, or character sequences [6]. Unlike speaker recognition and verification, the latter attempts to identify or confirm the speaker who uttered the voice rather than the vocabulary contained therein. In recent years, speech recognition technology has begun to change how we live and work and has gradually become a major man-machine interaction. The opening of this trend is due to the progress made in several key areas. First, Moore's Law continues to play a role [7]. Using multi-core processors, general-purpose image processing units, and CPU / GPU clusters, computing power has grown by several orders over 10 years. All this made it possible to train more powerful and complex models, and those models with more computational requirements significantly reduced the error rate of automatic speech recognition systems. Second, thanks to the rapid growth of the Internet and cloud computing, we can now get more data than ever. By building models from the vast amount of data collected from real-world scenarios, we can eliminate many of the assumptions made with the model and make the system more robust. [8-11].

3. Methodology

3.1. Voice Command Model Building

This section mainly focuses on the design of speech recognition instructions. Therefore, we first need to build a hidden Markov network model to complete all the instructions in the design process. The design process consists of two main steps: The first step is to match each factor with the Hidden Markov Network model and import it into memory for backup and storage, well prepared. Step two: It is necessary to connect the corresponding hidden Markov models to the hidden Markov networks in the order of pinyin structure through the Connect operation and design the silent mode to complete the construction of the hidden Markov network model. After completing the above two steps, we analyze the phoneme data in the hidden Markov network model and find that the data includes three different types: one is the state transition matrix of HMM data, including the state transition matrix Name, number of rows and columns, and specific data; second, the Gaussian mixture model data of the model phoneme data under different states, such as state name, Gaussian mixture number and Gaussian mixture model data of each Gaussian mixture; The third is Hidden Markov

Specific members	***Significance***
*char *transfer Name*	*The name of the transfer matrix*
*double **transfer data*	*Specific data of the transfer matrix*
*struct TRANSFER *next*	*Point to the next transfer matrix node*

Table 1. TRANSFER Specific Members and Meaning-Specific Members

Model data of phonemes, mainly including the names of hidden Markov models of phonemes, the number of states, the Gaussian mixture model of observation sequences output by each state, the transfer matrix data of all states, and the like.

The corresponding data are analyzed to construct the transfer matrix of the TRANSFER stored phoneme hidden Markov model data, and the Gaussian mixture model data of GAUSSIAN and STATE stored phoneme hidden Markov models under different states are sorted out. Then, the hidden Markov Model of HMM is built to store phonemes, and Table 1 is obtained. Due to the limited number of articles, this article does not elaborate on this.

The entire data storage process is encapsulated in the load Hmm (hmm FP) function, where hmm Fp is the phoneme-hidden Markov model file pointer obtained in the previous training module. Find T, find S, and find H, which are three numbers used to find the state transition matrix, state data, and hidden Markov model data. Define load Hmm

Complete the data storage. The '~ t' flag is then found in the data and stores the state transition matrix for all phonemic Hidden Markov Model data. Use the find T function to find the '~ t' flags. Each '~ t' flag represents a transition matrix, which stores the data for each transition matrix in the TRANSFER node, links all the TRANSFER nodes into a linked list, and gives the head of the transfer matrix linked list transfer Head. Find the '~ s' flag in the data and store the Gaussian mixture model data for each state in the phonemic Hidden Markov Model data. Use the find S function to find the '~ s' flags. Each '~ s' flag represents the state of a Hidden Markov Model, stores each state's data in sequence in the STATE node, links all the STATE nodes into a linked list, and gives the status of the head of the linked list state Head. Find the '~ h' flag in the data to store hidden Markov model data for phonemes. Use the find H function to find the '~ h' flag. Each 'h' flag represents hidden phone Markov model data for one phoneme. Each phoneme's Hidden Markov Model data is stored in the HMM node. All HMM nodes are linked into a linked list, and the head of the HMM model data list is given hmm Head.

Then, we construct a hidden Markov network for the entire instruction. We first analyze the single command hidden Markov chain construction process, "Please shut down" for example, "Please shut down" command has 6 phonemes, define the Connect operation, as shown in Figure 1, according to "Please turn off" Pinyin structure, The six phoneme Hidden Markov Models of _v-q + ing, c-ing + g, ing-g + uan, g-uan + j, uan-j + i, j-i + sil are sequentially connected in series Constructs a Hidden Markov Chain representing the "Please Shut Down" instruction. Two hidden Markov models connected; the last hidden Markov model deleted the last state, and a hidden Markov model deleted the first.

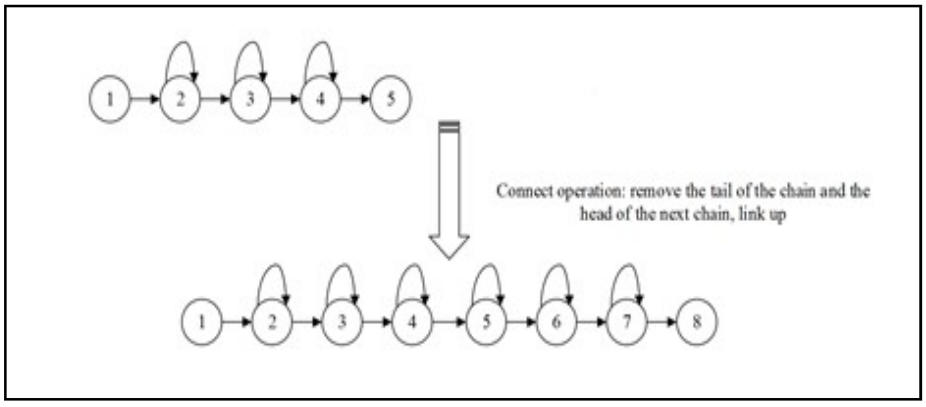

Figure 1. Connect operation

Then, define the Union operation, as shown in Figure 2, and concatenate each instruction's Hidden Markov Networks (Hidden Markov Networks) into a Hidden Markov Network. At the same time, the network uses the Connect operation plus silent mute Markov chains.

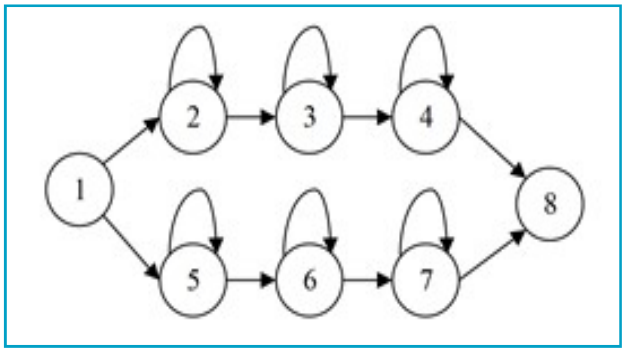

Figure 2. Union operation

Finally, according to the above analysis, we define the structure of NODE storage hidden Markov state nodes, LINK storage Markov chain, and NET storage hidden Markov network. We construct a single phoneme hidden Markov chain function hmm To Net function. Connect and Union operations are encapsulated in the connect Net and union Net, the entire network package is built in b Main Net, and the network model package function is free memory release. Using hmm To Net, the STATE of each phoneme is linked according to the hidden Markov data rule to a hidden Markov chain representing the phoneme. Use Connect Net to link the hidden Markov chains of each instruction's phonemes to form a hidden Markov chain representing the instruction. Using the union Net function, the hidden Markov chains of each instruction are linked side by side to form a hidden Markov network that represents all the instructions. Then, the Loop Print function was added to add hidden Markov network rotation; only with the rotation was the entire network completely constructed. Finally, a function for each network instruction is added to add an identification key. The key is the recognition result decoding, so it is very crucial.

3.2. Witt Algorithm Analysis

The token passing algorithm is obtained by improving the Viterbi frame synchronization decoding algorithm and is mainly used for searching non-jumping left-to-right hidden Markov model structures. In the search process, due to the different states of each data, resulting in a probability value, even if the same state of the data in the process of rotation also produces a corresponding probability value, in addition to the matching sequence probability has numerical values ?? are the key factor in decoding must be stored completely, so Token is treated as a storage of all probability values. At the beginning of execution, all different state nodes are initialized as a Token. With time changing, each frame of speech is continually processed, and the token shifts in the arc adjacent to it in the model until it reaches a state with an output probability density function and reaches—the final exit status. When a state node produces multiple state nodes, the Token will continue to follow the original route to spread so that the Token can continue to spread through all possible routes.

When the token is passed between different states, the probability cumulative value stored in the newly generated token is added with the probability value of the state transition and the probability value of the observation state corresponding to the new state. Still, the state may also rotate, and the state is updated Token. The cumulative value of the probability stored in the token is added with the probability of state rotation and the probability value of the observation state corresponding to the current state. Token walking in the network must record its path and voice instruction identifier. The degree of detail of the path reservation depends on the need to identify the output. In this design, we mainly record the log-likelihood and the voice command identifier. Find the one with the largest log-likelihood in the output node.

The token can return to its corresponding history record, find out all the state nodes and transfer arcs this Token is experiencing, and directly obtain the recognition result by using the voice instruction identifier in the Token data structure. For some state i at time t, we may get multiple tokens passed from t-1. How do you calculate the logarithmic probability of tokens in state j? In practice, we copy each token passing state i to its adjoining state j and use the value of (1) to increase the logarithmic probability of the token. The state updates itself at each moment Token.

$$\log[a_{ij}]+\log[b_j(o(t))] \quad (1)$$

Token transfer algorithm-specific implementation steps are as follows: the first is initialised, a token is set in the initial state of the hidden Markov network model, the initial probability value of the token is 0, and all other states set the token initial probability value ". This is followed by the loop iteration, starting from t = 1, the eigenvector of each frame signal processing, until the last frame signal; state jump process, the current state of the token copy will be passed to the new state, the new state token The probability cumulative value needs to be updated, assuming that the current state is i and the new state is j, the formula for the probability cumulative value in the new token is 2;

$$p_j(t)=p_i(t)+\log[a_{ij}]+\log[b_j(o(t))] \quad (2)$$

4. Result Analysis and Discussion

In the case of rotation, the calculation process is similar to equation (2), and state i rotates. Then, the probability accumulative value of the latest token of state i can be replaced by i of equation (2). Each state only keeps the token with the highest probability cumulative value. The token that checks all the ending states of the network model is terminated. We need the token with the highest probability cumulative value, and we can get the recognition result based on this token.

The following speech recognition for experimental testing was randomly selected from the sample of 14 test instructions, the instructions recorded in Table 2 below. The number of people involved in the test was 20, of which 10 were male and female, meeting the criteria for pronunciation in Putonghua. In this case, let each speaker read 14 test instructions at a speed of around 240 syllables per minute, requiring a pause of at least one second between the two instructions. Finally, a total of 280 voice test samples were collected. During the test, the environment must be kept quiet and environmental noise-free. Name each voice test sample according to the instruction name; for example, the first shutdown instruction will be named shutdown 1.

Please turn off the machine	Please open the machine	Please play video	Please open the address book	Please open the Notepad
Please take a picture.				Please turn off the video
Please turn off the alarm clock.	Please lock the screen	Please check the weather	Please open the browser	

Table 2. Speech Test Instructions

In this paper, noise is added to each test speech sample during the test to test the system's noise immunity. The noise used is additive white noise. During the test, four white noises of 15d B, 20d B, 25d B, and 30d B were added, and the above process was repeated. Table 3 lists the system in different signal-to-noise ratio environments, including all the voice commands to identify the accuracy. Figure 3 lists the different signal-to-noise ratio environment recognition rate curves.

By analyzing the trend changes in Figure 3 and the data in Table 3, we can see that the signal-to-noise ratio value gradually decreases, and the probability of the voice recognition system succeeding in recognition decreases during the noise increase. However, the recognition rate of the system selected in this paper The rate of decline relative to the speech recognition system slowed down. The minimum rate of decline in the recognition rate of this

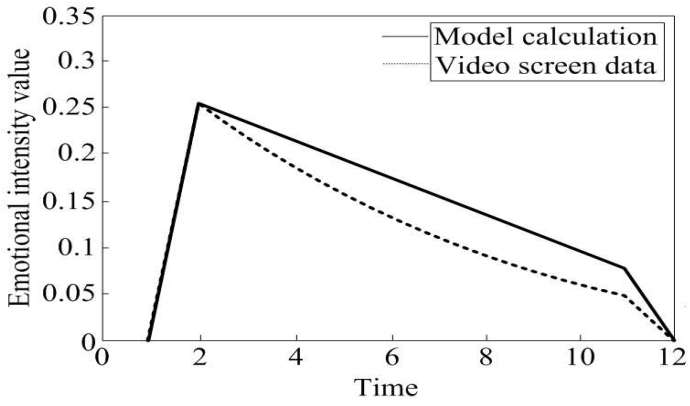

Figure 3. System recognition rate curve in different signal to noise ratio environment

Signal-to-noise ratio	Correct recognition of the number of voices	Total test speech number	Recognition rate
30dB	252	280	90.0%
21dB	264	280	94.2%
26dB	269	280	96.1%
35dB	274	280	97.8%

Table 3. Comparison of Recognition Rates of Different SNR Noises in Each System

system is in the SNR range of 20dB to 30 dB. When the SNR is lower than 20dB, the rate of decline of system identification starts to accelerate. In a word, this system has an anti-noise solid function. The analysis is mainly because this system training sample comes with noise,

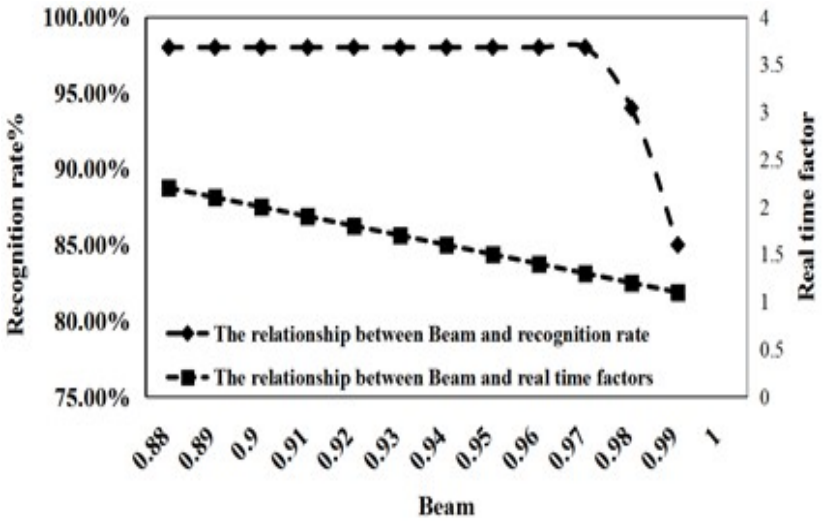

Figure 4. The curve of the recognition rate and the real-time factor of the system under different Beam

so the model has more noise immunity. From this point of view, we can choose samples with noises during training to improve speech system recognition rate and anti-interference and take this into full consideration in product design and development.

In the decoding process, the token that is too small is discarded, and a Beam is defined. The value of Beam is the ratio of the sum of the maximum token probability to the sum of the current token probabilities and obtains the value of Beam experimentally. The recognition rate and real-time factor obtained in Beam are as shown.

It can be inferred from Fig. 4 that the recognition rate changes as the Beam value increases. When the Beam value reaches 0.97, if the Beam value continues to increase, the recognition rate will decrease sharply, while the real-time factor will always follow the Beam value Increase continuously smaller. That is to say, as the beam value increases, recognition speeds up and faster. Considering the above data, if the system requires a high recognition rate, Beam is 0.97, corresponding to a recognition rate of 98.92% and a real-time factor of 0.38. If the system does not require a high recognition rate and requires higher speed, Beam can make 0.98, the corresponding recognition rate of 94.29%, a real-time factor of 0.22. The real-time factors 0.22 and 0.38 did not differ much, and the recognition speed is very small; considering the recognition rate and real-time factor, selecting Beam 0.97 is more appropriate. Table 4 compares the results of Beam with the 0.97 pruning algorithm and the initial token passing algorithm.

Type of test	Correct recognition number	Test sample number	Average recognition rate
Token passing	295	300	98.92%
Prune	299	302	98.92%

Table 4. Comparison of Test Results

As seen from the data analysis in Table 4, the pruning method can greatly enhance the decoding speed and improve the decoding efficiency. It is eight times faster than the original token transfer algorithm, and under the same recognition rate, it Improves the decoding level. In summary, this system uses a Beam of 0.97 pruning decoding.

5. Conclusion

While intelligent technology gradually covers people's lives and work, speech recognition technology has developed rapidly and plays an important role in the interaction between humans and machines. It has also become an important method for training foreign language personnel. In the upcoming 2022 Winter Olympic Games held in our country, in the face of participating countries and players worldwide, the selection and training of foreign language talent is significant. Therefore, research on speech recognition technology in foreign language talent cultivation is an urgent task. In this context, this paper used the Viterbi algorithm further to explore the application of foreign language talent training. Based on the actual situation of the training of foreign language talents in our country, the paper constructed the voice instruction model based on the related theory, improved the Viterbi algorithm, and verified it through experiments to ensure the advantages and feasibility of the improved Viterbi algorithm.

References

[1] Ailing, Q. (2017). A study on college English autonomous learning model based on ESP theory. *Agro Food Industry Hi Tech, 28*(1), 904-907.

[2] Zhang, J. (2013). The ESP instruction: A study based on the pattern of autonomous inquiry. *English Language Teaching, 6*(3), 12-16.

[3] Chen, J. (2016). Interactive approach to teaching ESP reading in the autonomous learning classroom—A corpus-based discourse information analysis. *Foreign Language World, 6*(1), 26-29.

[4] Liu, X. (2016). Research into the influence of Internet-based ESP teaching and learning model on learner autonomy. *Foreign Language Research, 10*(1), 12-16.

[5] Ajideh, P. (2016). Autonomous learning and metacognitive strategies essentials in ESP class. *English Language Teaching, 2*(1), 162.

[6] Wang, F. Y. (2015). Study on establishment of English corpus of higher vocational schools based on ESP teaching—The case of Huzhou Vocational and Technical College. *Vocational & Technical Education, 2015*(2), 12-16.

[7] Hüttner, J., Smit, U., Mehlmauer-Larcher, B. (2016). ESP teacher education at the interface of theory and practice: Introducing a model of mediated corpus-based genre analysis. *System, 37*(1), 99-109.

[8] Tandel, N. H., Prajapati, H. B., Dabhi, V. K. (2020). Voice recognition and voice comparison using machine learning techniques: A survey. In *2020 6th International Conference on Advanced Computing and Communication Systems (ICACCS)* (pp. 459-465). IEEE. https://doi.org/10.1109/ICACCS48705.2020.9074184.

[9] Singh, N., Agrawal, A., Khan, R. A. (2017). Automatic speaker recognition: Current approaches and progress in last six decades. *Global Journal of Enterprise Information System*, 9(3), 45-52.

[10] Dong, Y. (2022). Application of artificial intelligence software based on semantic web technology in English learning and teaching. *Journal of Internet Technology*, 23(1), 143-152.

[11] Dizon, G., Tang, D. (2020). Intelligent personal assistants for autonomous second language learning: An investigation of Alexa. *JALT CALL Journal*, 16(2), 107-120.