Eng Int Syst (2025) 4: 397–404
© 2025 CRL Publishing Ltd

Engineering
Intelligent Systems

Implementation of Voice Software Testing Framework for Intelligent AI Technology

Chao lv*

Faculty of Computer Science, Haojing College of Shaanxi University of Science & Technology, Xi'an 712000, Shaanxi, China

The voice software testing framework is an information-based testing method aimed at verifying the functionality, reliability, and performance of voice applications, and plays a crucial role in the testing of various voice applications. The main role of testing frameworks is to provide high-quality and stable testing, and to help developers better understand the quality and performance of voice applications. Artificial intelligence (AI) technology plays an important role in voice software testing frameworks. By utilizing AI technology, automated testing can be achieved to improve test coverage and efficiency and identify more potential problems. In this project, the voice software testing framework was studied through AI technology. By conducting comparative experiments, the traditional Hidden Markov Model (HMM) and algorithms based on AI technology were compared to evaluate their performance. After experimental verification, it was found that in the field of voice recognition, the average recognition speeds of traditional HMM algorithms and voice recognition judgment algorithms based on AI technology for Chinese, English, and emotional audio were 579ms, 568ms, 623ms, and 533ms, 526ms, and 589ms, respectively. Experimental results showed that voice recognition judgment algorithms based on A technology performed better in terms of recognition speed for different audio types. In addition, this algorithm also outperformed traditional HMM algorithms in terms of recognition error rates for various audio types, indicating that it has stronger audio recognition capabilities. The excellent performance of voice recognition judgment algorithms based on AI technology has been demonstrated through experiments, providing a new direction for the design and research of voice software testing frameworks.

Keywords: Voice Software, Artificial Intelligence, Test Framework, Voice Recognition and Judgment Algorithm

1. INTRODUCTION

With the rapid development of voice technology, voice testing has become increasingly important. The performance and quality of voice technology are crucial for user experience in various application scenarios. Hence, voice testing has become an indispensable part of the development of voice technology. With the rapid development and popularization of intelligent AI technology, voice testing can achieve automation, efficiency, and better test coverage, thereby improving the quality and performance of voice applications and further improving the testing experience.

The literature review yielded a substantial amount of research material, the most relevant of which are described here. Bernstein believed that the ability to recognize words in connected voice was crucial for daily communication in noisy listening conditions. The study used various open set recognition error mining methods in multi project testing to obtain different ability measures. She demonstrated that in noise testing, error mining methods using open set responses of clinical sentences could be used to characterize the ability to exceed signal-to-noise ratio thresholds. This study also used a stimulus response phoneme to phoneme sequence alignment software system to achieve automatic and accurate quantitative error scoring, and evaluated the relationship between two types of answer errors and word correct scores using mixed model regression [1]. Donhauser explored how the human brain used contextual prediction to optimize sensory sampling and processing during voice listening, and

*Corresponding author. E-mail: hjlvc@126.com

investigated whether this prediction processing was dynamically organized into separate oscillatory time scales. He used neural network technology to predict voice at the phoneme level using context. Through this model, he estimated the contextual uncertainty of natural language and explained the factors affecting the neurophysiological activities of human listeners [2]. Ault conducted research on voice recognition algorithms and pointed out that voice recognition algorithms are now a part of people's daily lives because they are used for various tasks, such as converting sound waves into useful data that machines can process and interpret. He believed that neural networks were replacing Gaussian mixture models as the main method for converting voice into text. Neural networks are more effective in generating correct output and can better understand input by comparing words and using memory, thereby analyzing more complex sentences. He explored multiple directions in the field of voice testing and provided useful guidance for related research [3]. Haridas believed that human voice recognition had always been a concern for AI and processing researchers. Therefore, his research provided an overview of voice recognition strategies applicable to human recognition. He explored existing voice recognition strategies and overcame the limitations of existing indicators. In addition, he distinguished various issues included in voice recognition methods and studied different voice recognition programs, providing a detailed introduction to voice recognition technology [4]. It can be seen that these articles have conducted detailed research on voice testing, providing valuable information for this article.

The study conducted by de la Fuente Garcia focused on the monitoring of Alzheimer's disease using AI, voice, and language processing methods. She believed that language provided a valuable source of clinical information in Alzheimer's disease. She summarized existing findings on the use of AI, voice, and language processing to predict cognitive decline in patients who had Alzheimer's disease. Moreover, de la Fuente Garcia explained current research procedures, recognized their limitations, and proposed strategies to address them [5]. Zhu used bibliometric software to analyze the research status and current topics of interest of AI in the fields of intelligent voice processing and social sciences. Secondly, he explored the current research status of AI in the fields of intelligent voice processing and social sciences, and explored the current hotspots of AI research through keyword co-occurrence networks and cluster analysis. In addition, he also discussed possible future research directions [6]. Although these researchers have contributed valuable ideas and insights, their focus has been mainly theory-related, and without empirical verification.

In this current study, a voice software testing framework was designed through a voice recognition judgment algorithm based on AI technology. According to the experimental results, for Chinese audio, the average sentence error rate (SER), word error rate (WER), and character error rate (CER) of the HMM algorithm were 4.596%, 5.136%, and 5.585%, respectively. On the other hand, the average SER, WER, and CER of the AI- based algorithm were 4.033%, 4.416%, and 4.708%, respectively. Similarly, for English audio, the average SER, WER, and CER of the HMM algorithm were 3.952%, 4.488%, and 4.770%, respectively, while the average SER, WER, and CER of algorithms based on AI were 3.015%, 3.712%, and 4.407%, respectively. It can be seen that AI-based algorithms have more significant advantages in in terms of language recognition performance. The innovative contribution made by this current study the application of AI technology to optimize the voice software testing framework.

2. VOICE TESTING TECHNOLOGY

2.1 Main Process of Voice Testing

Voice software testing refers to the process of testing voice technology software such as voice recognition, voice synthesis, and natural language processing [7–9]. Its purpose is to ensure that voice software can accurately recognize and process voice information in various situations, thereby improving its reliability and stability. In addition, voice software testing is a constantly evolving and improving process that requires continuous updates of test cases and methods so as to adapt to new technologies and requirements. At the same time, it is also necessary to analyze and provide feedback on the test results in order to optimize the voice software so as to improve its performance and reliability. The testing of voice software comprises the following main steps.

(1) Environmental preparation

In voice software testing, environmental preparation is a very important part [10–11]. Because voice testing requires a series of operations such as collecting, cleaning, and testing voice data, the stability and accuracy of the environment directly determine the quality of the test data [12–13]. Specifically, during the environmental preparation process, the following points need to be noted:

Environment selection: Choosing a quiet and undisturbed environment is very important. It is best to choose a closed room or soundproof room where there are no other people or objects interfering with the testing process. At the same time, attention should also be paid to avoiding environmental noise, such as air conditioning noise, mechanical noise, etc.

Microphone selection: The choice of microphone is also very important. It is necessary to choose high-quality microphones that can capture clear and accurate voice signals and eliminate the influence of environmental noise. Commonly used microphone types include capacitive microphones, dynamic microphones, and directional microphones.

(2) Building test sets

According to the different testing tasks, it is necessary to select a suitable voice dataset and perform preprocessing, annotation, and other work to construct a test set [14–15]. Test sets usually require a wide range of high-quality data, while also meeting the requirements of testing tasks and indicators. Its purpose is to evaluate the performance and accuracy of phonetic algorithm or model. Building a test set generally requires consideration of the following aspects:

Voice dataset selection: Firstly, the voice dataset suitable for the testing task needs to be selected. For example, if the testing task is voice recognition, it is possible to consider using annotated voice corresponding text datasets, such as LibriSpeech, zhddline, and Mozilla Common Voice. If the testing task is speaker recognition, it is necessary to select

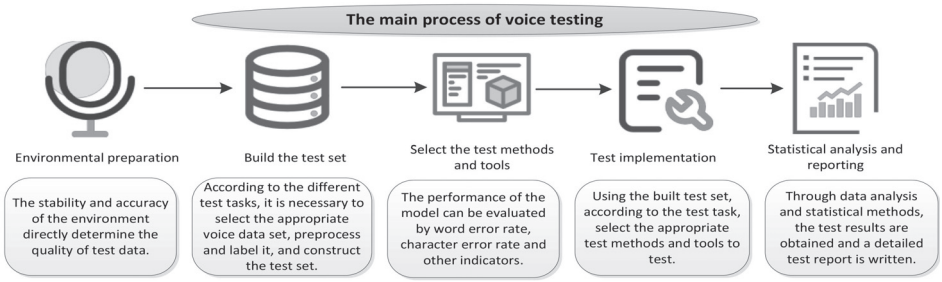


Figure 1 Voice-testing flowchart.

a voice dataset of speakers with different ages, genders, backgrounds, etc.

Data preprocessing: When constructing a test set, it is necessary to preprocess the original voice data by removing noise, improving sound quality, and simplifying data. For voice recognition tasks, it is also necessary to perform operations such as framing and feature extraction on the voice signal, in order to facilitate subsequent feature engineering and model training.

Data annotation: The voice data in the test set needs to be annotated to facilitate the calculation and evaluation of test indicators. For different testing tasks, annotation can involve text transcription, various features such as pronunciation, emphasis, tone, etc.

Coverage: The test set needs to cover as many aspects of voice data as possible, such as the speaker's age, gender, accent, pronunciation style, speed, and other features of speech, in order to test the generalizability of algorithms or models.

(3) Selecting testing methods and tools

Appropriate testing methods and tools are selected. For example, in voice recognition testing tasks, indicators such as sentence SER, WER, and CER can be used to evaluate the performance of the model. These three indicators can indicate the error rate of the model in order to evaluate the accuracy of the model. In general, the lower the SER, WER and CER, the greater is the recognition accuracy of the system.

SER is an evaluation metric commonly utilized in voice recognition test tasks to assess the performance of automatic voice recognition systems, and determines the percentage of sentences in a test dataset that are recognized with errors. A lower SER indicates a better performance of the automatic voice recognition system. For example, if an original text contains 100 sentences and the voice recognition system outputs a text with 5 incorrect sentences, the SER is 5%.

WER is the percentage of the total number of words in the original text that the voice recognition system processes in a piece of audio text that does not agree with the original text. For example, if an original text contains 100 words, and the voice recognition system outputs text with 8 incorrect words, the WER is 8%.

CER is the percentage of characters in the original text that do not match the original text when the voice recognition system is processing a piece of audio. CER is more granular than WER and reflects the accuracy of each character. For example, if an original text contains 1000 characters and the speech recognition system outputs 20 characters that are incorrect, then the CER is 2%.

(4) Test implementation

Tests are performed using a constructed test set, selecting appropriate test methods and tools based on test tasks, metrics, and methodology. The testing process may require operations such as adjusting parameters and optimizing models to achieve better performance. When performing language testing, the following steps usually need to be implemented:

Selecting suitable testing tools: According to the testing tasks, indicators, and methods, suitable testing tools are selected. Operations such as feature extraction and acoustic model training are performed on the speech signal.

Testing: After preparing the test dataset and corresponding testing methods and tools, the tests can commence. Operations such as adjusting parameters, optimizing models, etc. may be required during the testing process to achieve better performance.

(5) Statistical analysis and reporting

Data analysis and statistical methods are utilized to derive test results and write detailed test reports, including test methods, data analysis results, and recommendations and improvement measures. Among them, the following steps are included:

Data analysis: The test outcomes are processed and analyzed, and statistical methods are utilized to obtain relevant indicators, such as average score, standard deviation, ranking, etc. Furthermore, the reasons behind the test results can be explored in depth with methods such as cross-tabulation analysis.

Writing a test report: Based on the test objectives and data analysis results, a detailed test report is written. The report includes testing objectives, testing methods, data analysis results, testing conclusions, as well as suggestions and improvement measures. The test report needs to be concise, logical, readable, and ensure the privacy protection of the test subject.

The voice-testing process is shown in Figure 1.

2.2 Main Structure of Voice Software Testing Framework

The origin of voice software testing frameworks can be traced back to the origin of computer software testing. In the early stages of computer software development, software testing was mainly done manually. Testers needed to test individual test cases, which resulted in low efficiency and inconsistent testing quality [16–18]. With the development and increasing complexity of computer technology, software

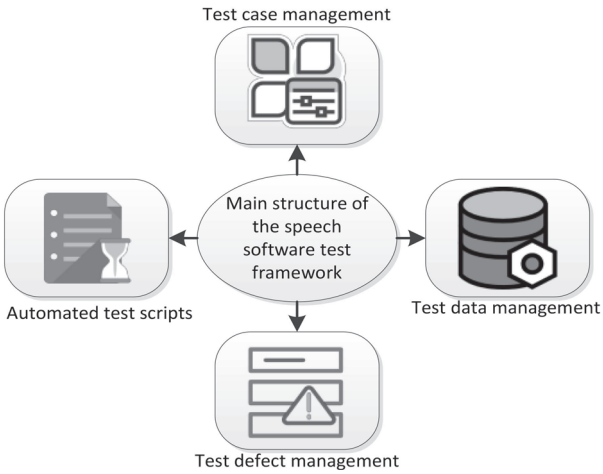


Figure 2 Diagram of main structure of voice software testing framework.

testing has gradually become an important task. Testers need to adopt more effective testing methods and tools to improve testing efficiency and quality. The main structure of a testing framework typically includes the following:

(1) Test case management

Test case management is an important component of the voice software testing framework, and its main function is to version control and manage test cases [19]. In the process of software testing, test case libraries are usually created and maintained by testers or testing teams, containing various types of test cases, such as normal scenario testing, abnormal scenario testing, boundary testing, etc.

In test case management, version control is required for test cases to track their evolution history and restore previous versions during testing. Corresponding interfaces need to be provided to support batch execution of test cases and generation of test reports. Through these interfaces, testers can easily execute test cases and generate test reports, while also optimizing and adjusting test results.

(2) Automated test scripts

An automated test script is a program that automatically simulates manual operations in order to test the functionality, performance, etc., of a software system. The voice testing framework is one that is used for testing a certain programming language. In order to achieve automated testing of software systems, testers need to write programmable test scripts.

Testers can use these test scripts to simulate various operations, such as launching applications, inputting data, clicking buttons, etc., to automate the testing of software systems. Through automated testing, testing efficiency can be improved, testing costs can be reduced, and problems in software systems can be identified more quickly. Therefore, test scripting tools are an important part of automated testing. Programmability, ease of use, and compatibility of test scripting tools need to be considered when selecting a testing framework so that testers can write high-quality, reliable automated test scripts.

(3) Test data management

Test data management refers to the collection, storage, use, and management of test data during the software testing process. Test data is the foundation of test case execution; hence, the quality of test data management directly affects the effectiveness of software testing. Usually, test data needs to be prepared in advance and stored in the corresponding data warehouse. The testing framework needs to provide a data access interface, so that test cases can read test data for testing operations. In order to ensure the reliability and consistency of test data, the following aspects should also be considered for test data management:

Version management of data: Test data may need to be managed and updated according to different test versions, so corresponding version management mechanisms are needed.

Data security: The test data may contain sensitive information such as passwords, so it is necessary to consider the security of the test data and set corresponding permissions and access control mechanisms [20–21].

Data reuse: Generally speaking, test data can be shared by multiple test cases. Therefore, it is necessary to support data reuse to improve testing efficiency.

(4) Test defect management

When voice testing, testers usually discover various problems and defects that need to be recorded and managed in a timely manner. Through defect management, testing teams are better able to handle defects, timely identify and solve problems in software systems, and improve the quality and stability of software systems. Therefore, the testing framework needs to provide defect management functions, including the following.

Defect submission: Testers can submit defects through the testing framework and save relevant information of defects (such as defect description, recurrence steps, priority, severity, etc.) to the defect library.

Defect tracking: Defect tracking refers to tracking and recording the status and handling of defects, in order to timely understand the handling of defects.

Defect resolution: Testers can mark resolved defects as closed through the testing framework, and the resolution method and processing process are recorded in the defect library.

The main structure of the voice software testing framework is shown in Figure 2.

3. AI TECHNOLOGY

3.1 Role of AI Technology in Voice Software Testing

AI technology can play an important role in voice software testing. First of all, taking natural language processing technology as an example, during software testing, by analyzing text data and using deep learning and other technologies to train models, the software can better understand and process the language used by humans, thus improving the accuracy and efficiency of software translation, grammar detection, text matching and other tasks. Secondly, AI technology can also help testers analyze and predict user language. According to the user's language characteristics, test cases are automatically generated to improve the comprehensiveness and specificity of test cases, while also optimizing the testing process and shortening testing time.

In addition, AI technology can also significantly reduce testing costs and improve the quality and stability of testing results by simulating user behavior, automating testing processes. In short, the application of AI technology in voice software testing can help enterprises save costs, shorten development cycles, improve product quality, and reduce research and development risks, thus promoting innovation and the development of enterprises.

3.2 Voice Recognition Judgment Algorithm Based on AI Technology

Voice recognition methods based on statistical models are dominant in voice recognition research. The basic principle is: After the voice signal is processed in the front end, the acoustic feature sequence O is obtained, and the maximum posterior probability probability criterion is used in the decoder to find an optimal word sequence W, as shown below:

$$\widehat{W} = \max(p(W|O)) = \max \frac{p(W)p(o|W)}{p(O)} \tag{1}$$

where $p(W)$ is the prior probability of the occurrence of a specific word sequence. $p(O|W)$ is the probability of outputting O given a word sequence of W. The above formula can be converted to:

$$\widehat{W} = \max(p(W)p(O|W)) \tag{2}$$

Then, cross-entropy is used to test the performance of voice recognition algorithm. For language model M, test set T consists of sentence sequence $t_1 t_1 \ldots t_n$. The expression of its cross-entropy is as follows:

$$H(P_T, P_M) = -\frac{1}{N_T} \sum_{i=1}^{n} \log_2 P_M(t_n) \tag{3}$$

where n is the number of sentences contained in test set T. N_T is the total number of words in test set T. $P_M(t_n)$ is the probability calculated by model M for sentence t_n. Secondly, the perplexity of the algorithm is calculated, and its formula is as follows:

$$C = 2^{H(P_T, P_M)} \tag{4}$$

Finally, the word error rate is calculated, and its expression is as follows:

$$WR = \frac{D + I + S}{N} \tag{5}$$

where N is the number of all words; D is the number of deletion errors; I is the number of word insertion errors; and S is the number of substitution errors. With this algorithm, voice audio can be effectively recognized and judged, providing strong support for the realization of intelligent voice testing.

4. EXPERIMENT ON VOICE SOFTWARE TESTING FRAMEWORK BASED ON AI TECHNOLOGY

4.1 Basic Explanation of the Experiment

In the experiments, the performance of speech recognition judgment algorithms based on AI techniques was tested. For different speech recognition algorithms, different performance metrics can be used to evaluate their performance. To objectively evaluate the performance of voice recognition AI-based algorithms, comparative experiments are usually performed. The traditional HMM is also a very classical recognition algorithm in the voice recognition field. Therefore, in this study, the HMM algorithm was selected for comparison with the algorithms based on AI techniques. By comparing the test results, the performance indicators of different algorithms can be obtained to evaluate their accuracy and efficiency.

4.2 Selection of Experimental Datasets

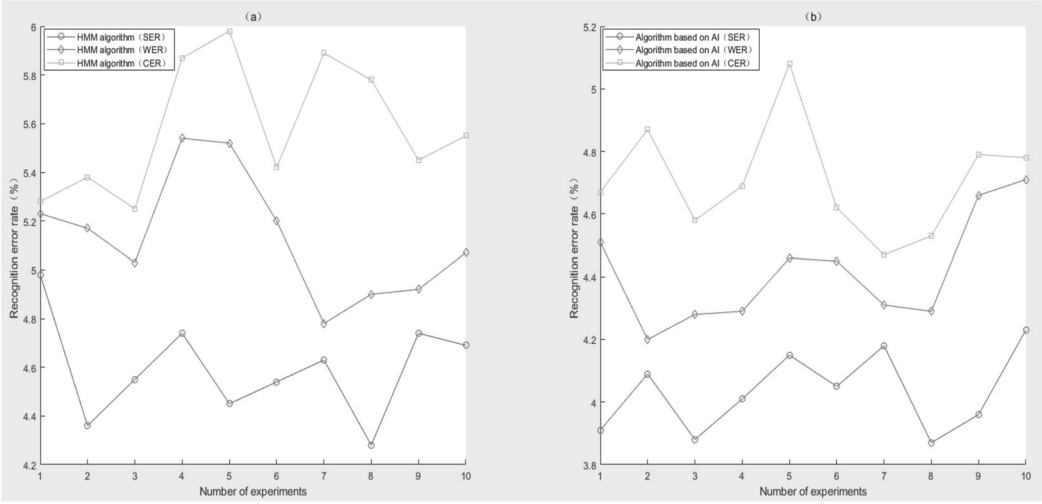
For voice test experiments, the selection of a suitable test speech dataset is vital because it directly affects the results of the test as well as the difficulty and complexity of the test task. Different testing tasks need different types of voice datasets. Conventional voice datasets usually use labeled speech-corresponding text datasets such as LibriSpeech, zhddline, and Mozilla Common Voice, which contain numerous voice files and corresponding text files that can be used to train and test voice recognition models. LibriSpeech is used mainly for English voice recognition tasks, and zhddline is used mainly for Chinese voice recognition tasks. Mozilla Common Voice is a multilingual dataset that covers over 20 languages of voice data and also has multiple types of voice emotion data.

4.3 Selection of Experimental Audio Types

Chinese Voice recognition: Chinese is one of the most complex languages in the world comprising Chinese characters, pinyin, rare characters, dialects, and other language features. Moreover, Chinese grammar and expression are quite complex. For example, a word may have multiple different meanings and interpretations, and its specific meaning needs to be determined through context.

Table 1 Average recognition speed of different algorithms.

Audio type	Audio quantity	Traditional HMM algorithm	Voice recognition and judgment algorithm based on AI technology
Chinese	3000	579 ms	533 ms
English	2000	568 ms	526 ms
Feeling	500	623 ms	589 ms



(a) Recognition error rate of Chinese audio based on HMM algorithm.

(b) Recognition error rate of Chinese audio based on AI algorithm.

Figure 3 Recognition error rate of Chinese audio using different algorithms.

English speech recognition: English is one of the most widely used languages in the world, characterized by simple grammar and a rich vocabulary. The processing of English information is less difficult because the vocabulary and expression in English are relatively stable, and its grammar is relatively simple, which is easy for natural language processing.

Voice emotion recognition: This type of audio data is used to recognize the speaker's emotional state. When people speak, they not only transmit information through language; they also convey their emotional states such as happiness, anger, sadness, etc. through tone, Therefore, the purpose of voice emotion recognition is to extract these emotional information from voice signals, in order to better understand the speaker's true emotional state.

4.4 Selection of Experimental Indicators

Firstly, in language testing experiments, the recognition speed of the algorithm for audio is very important, and this indicator can reflect the performance and practicality of the algorithm. Specifically, if an algorithm has a relatively fast recognition speed under different audio types, then this algorithm has good practical application.

In addition, in this voice recognition test, the performance of the voice recognition model can be evaluated through other indicators such as SER, WER, and CER. These three indicators can reflect the language recognition error rate of the model, so they can be used to evaluate the accuracy of the model.

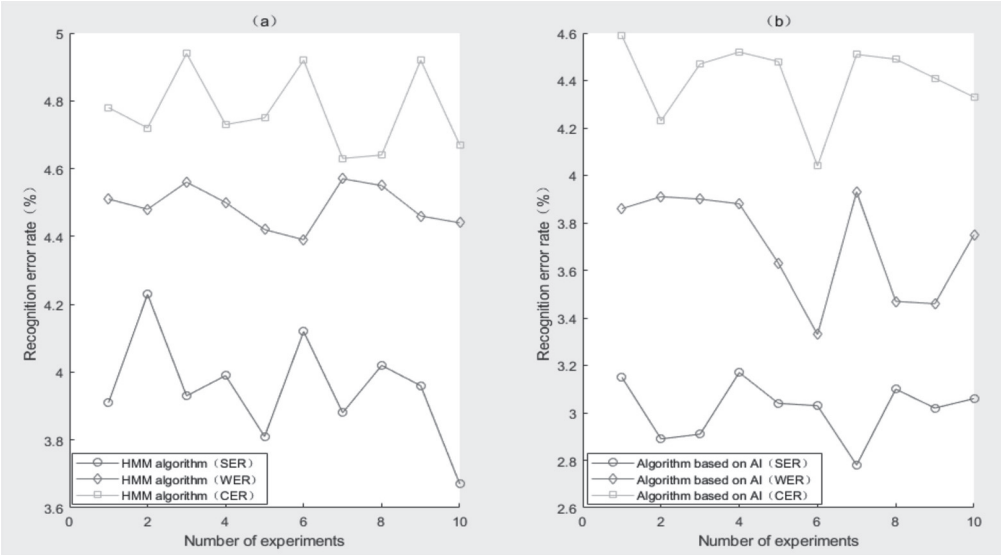
4.5 Experimental Results

Firstly, the recognition speed of different algorithms for different audio types was tested. The specific results are shown in Table 1.

The experimental results shown in Table 1 indicate that the average recognition speed of traditional HMM algorithms for Chinese, English, and emotions is 579 ms, 568 ms, and 623 ms, respectively. The average recognition speeds of voice recognition algorithms based on AI technology for Chinese, English, and emotions are 533 ms, 526 ms, and 589 ms, respectively. Obviously, voice recognition algorithms based on AI technology demonstrate faster recognition speeds across different audio types.

Secondly, the recognition error rate data of different algorithms for Chinese audio were tested and displayed through indicators such as SER, WER, and CER. The experimental data is shown in Figure 3.

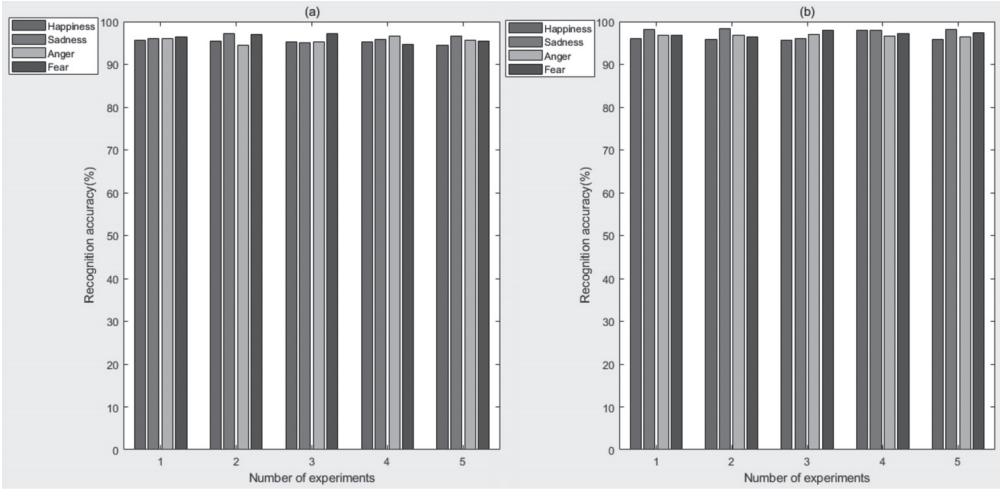
Figures 3 (a) and 3 (b) indicate that the average SER, WER, and CER of the HMM algorithm for Chinese audio were 4.596%, 5.136%, and 5.585%, respectively. The average SER, WER, and CER of Chinese audio based on AI algorithms were 4.033%, 4.416%, and 4.708%, respectively. This indicates that the algorithm has a lower recognition error rate for Chinese audio.



(a) Recognition error rate of English audio based on HMM algorithm.

(b) Recognition error rate of English audio based on AI algorithm.

Figure 4 Recognition error rate of English audio using different algorithms.



(a) Accuracy of emotion recognition based on HMM algorithm, Figure 5

(b) Accuracy of emotion recognition based on AI algorithm.

Figure 5 Accuracy of emotion recognition using different algorithms.

Secondly, the recognition error rate data of different algorithms for English audio were tested and displayed through indicators such as SER, WER, and CER. The experimental data is shown in Figure 4.

Figures 4 (a) and 4 (b) show that the average SER, WER, and CER of the HMM algorithm for English audio were 3.952%, 4.488%, and 4.770%, respectively. The average SER, WER, and CER of English audio based on AI algorithms were 3.015%, 3.712%, and 4.407%, respectively. This indicates that the algorithm has a lower recognition error rate for English audio.

Finally, the accuracy of different algorithms in emotion recognition was tested. In the experiment, happiness, sadness, anger, and fear were tested. The experimental results are shown in Figure 5.

Figures 5(a) and 5(b) show that the average recognition accuracy of the HMM algorithm for happiness, sadness, anger, and fear is 95.204%, 96.190%, 95.642%, and 96.134%, respectively. The average recognition accuracy of AI-based algorithms for happiness, sadness, anger, and fear is 96.240%, 97.676%, 96.752%, and 97.196%, respectively. It can be seen that the AI-based algorithm demonstrates a higher recognition accuracy for various emotions.

5. CONCLUSIONS

Voice software testing technology is a widespread and constantly evolving field. In practical applications, the testing of voice software can help developers improve the quality of such software, eliminate defects, and improve user experience. The voice software testing framework based on AI is an emerging testing method. The core idea is to verify the stability and effectiveness of voice software

in various usage scenarios by simulating voice input in real scenarios. This testing framework can reflect the actual application of voice software more comprehensively and improve the accuracy and validity of the test results. The experimental results obtained in this study indicate that the voice recognition judgment algorithm based on AI technology has faster language recognition speed and more accurate audio recognition accuracy, while having excellent performance in the recognition of emotion. In conclusion, the AI-based voice software testing framework is an advanced testing method with high accuracy and credibility, which has an increasing number of applications in the software-testing field in future. However, there are still several issues that research needs to address. For example, the classification and recognition of voice signals also require attention to factors such as slang, dialects, and accents, which may make incorrect classification decisions for certain specific sounds or accents, thereby affecting test results. In addition, there are scalability issues, and the application of AI technology in testing frameworks needs to be constantly updated and improved. However, if the testing framework itself lacks scalability, then with the development of technology, the testing framework may become outdated. Therefore, these issues need to be fully taken into account in future studies and corresponding measures must be taken to address them.

REFERENCES

1. Bernstein, Lynne E., Silvio P. Eberhardt, and Edward T. Auer Jr. “Errors on a speech-in-babble sentence recognition test reveal individual differences in acoustic phonetic perception and babble misallocations”. Ear and Hearing 42.3 (2021): 673–690.
2. Donhauser, Peter W., and Sylvain Baillet. “Two distinct neural timescales for predictive speech processing”. Neuron 105.2 (2020): 385–393.
3. Ault, Shaun V., Rene J. Perez, Chloe A. Kimble, and Jin Wang. “On speech recognition algorithms”. International Journal of Machine Learning and Computing 8.6 (2018): 518–523.
4. Haridas, Arul Valiyavalappil, Ramalatha Marimuthu, and Vaazi Gangadharan Sivakumar. “A critical review and analysis on techniques of speech recognition: The road ahead”. International Journal of Knowledge-Based and Intelligent Engineering Systems 22.1 (2018): 39–57.
5. De La Fuente Garcia, Sofia, Craig W. Ritchie, and Saturnino Luz. “Artificial intelligence, speech, and language processing approaches to monitoring Alzheimer’s disease: a systematic review”. Journal of Alzheimer’s Disease 78.4 (2020): 1547–1574.
6. Zhu, Chengke, and Xiaofeng Zhao. “Research hotspots of current interest and trend of artificial intelligence in intelligent speech processing and social sciences–visualized comparison research based on CiteSpace”. Nanotechnology for Environmental Engineering 7.3 (2022): 843–855.
7. Benkerzaz, Saliha, Youssef Elmir, and Abdeslam Dennai. “A study on automatic speech recognition”. Journal of Information Technology Review 10.3 (2019): 77–85.
8. Trivedi, Ayushi,Navya Pant, Pinal Shah,Simran Sonik and Supriya Agrawal. “Speech to text and text to speech recognition systems-Areview.” IOSR Journal of Computer Engineering 20.2 (2018): 36–43.
9. Nicholson Sahil. “A Speech Remote Control System for Intelligent Robots by the Finite Volume Method”. Kinetic Mechanical Engineering, 1(1), (2020): 43–50.
10. Singh, Amitoj,Navkiran Kaur, Vinay Kukreja, Virender Kadyan, Munish Kumar. “Computational intelligence in processing of speech acoustics: a survey”. Complex & Intelligent Systems 8.3 (2022): 2623–2661.
11. Hassan, Muhammad D.,Ali Nejdet Nasret, Mohammed Rashad Baker, Zuhair Shakor Mahmood. “Enhancement automatic speech recognition by deep neural networks”. Periodicals of Engineering and Natural Sciences 9.4 (2021): 921–927.
12. Meng, F., Zheng, Y., Bao, S., Wang, J., & Yang, S. “Formulaic language identification model based on GCN fusing associated information”. Peer Computer Science, 8, (2022): e984.
13. Zhou, Y. & Bu, F. “An Overview of Advancements in Lie Detection Technology in Speech”. International Journal of Information Technologies and Systems Approach, 16(2), (2023): 1–24.
14. Shewalkar, Apeksha, Deepika Nyavanandi, and Simone A. Ludwig. “Performance evaluation of deep neural networks applied to speech recognition: RNN, LSTM and GRU”. Journal of Artificial Intelligence and Soft Computing Research 9.4 (2019): 235–245.
15. Liang, Sendong, and Wei Qi Yan. “A hybrid CTC+ Attention model based on end-to-end framework for multilingual speech recognition”. Multimedia Tools and Applications 81.28 (2022): 41295–41308.
16. Vryzas, Nikolaos, Vrysis, Lazaros; Matsiola, Maria; Kotsakis, Rigas; Dimoulas, Charalampos; Kalliris, George. “Continuous speech emotion recognition with convolutional neural networks”. Journal of the Audio Engineering Society 68.1/2 (2020): 14–24.
17. Kwon, Soonil. “Optimal feature selection based speech emotion recognition using two-stream deep convolutional neural network”. International Journal of Intelligent Systems 36.9 (2021): 5116–5135.
18. Meng Fanqi, Cheng Wenying, Wang Jingdong. “Semi-supervised Software Defect Prediction Model Based on Tri-training”. KSII Transactions on Internet and Information Systems, 15(11), (2021): 4028–4042.
19. Qin, Chu-Xiong, Wen-Lin Zhang, and Dan Qu. “A new joint CTC-attention-based speech recognition model with multi-level multi-head attention”. EURASIP Journal on Audio, Speech, and Music Processing 2019.1 (2019): 1–12.
20. J. Zhang, “Real-time Integration Technology of Large-scale Heterogeneous Data Based on Big Data and Artificial Intelligence”, Engineering Intelligent Systems, vol. 33 no. 2, pp. 179–188, 2025.
21. L. Hou, “Design of Function Approximation Algorithm Based on RBF Neural Network”, Engineering Intelligent Systems, vol. 33 no. 2, pp. 155–167, 2025.