Automated Interview Evaluation

Ch. Sri Latha¹*, N. Krishna Kalyan¹, J. Abhilash¹, O. V. S. N. Kaushik¹, Vinod Balmiki², S. Venukumar³

¹Department of AIMLE, GRIET, Hyderabad, Telangana State, India.
²Uttaranchal Institute of Technology, Uttaranchal University, Dehradun, 248007, India
³KG Reddy College of Engineering & Technology, Hyderabad, India

Abstract. The Interview Automation System is an Artificial Intelligence (AI) powered platform designed to help users prepare for real-life interviews. By asking questions and evaluating responses against original answers, our system provides users with a valuable opportunity to practice and improve their interview skills. The research aims to increase users' familiarity with the interview process, ultimately boosting their confidence and improving their chances of success. The workflow of the research involves several key steps, including data pre-processing and database management, audio-based question delivery, and audio-based answer collection and evaluation. Our platform is easily scalable and can be expanded to include additional functionalities such as users' confidence detection, resume-based topic questioning, and body language analysis.

1 Introduction

The interview preparation platform helps users prepare for job interviews by providing them with a comprehensive and interactive experience. The platform uses a variety of technologies, including speech recognition, machine learning, and natural language processing, to provide users with a seamless and user-friendly experience that helps them build their confidence and enhance their capabilities. The platform consists of several components, including a web interface, a database for storing questions and user responses, and several machine learning models for evaluating user performance. The web interface is designed to be intuitive and user-friendly, allowing users to easily navigate through the platform and interact with its various features. One of the key features of the platform is the use of speech recognition to capture user responses. This allows users to respond to interview questions in a conversational manner, just as they would in a real interview. The platform also uses machine learning models, such as the SBERT model, to evaluate the user's responses and provide them with feedback on their performance. The evaluated results of the user are then displayed in a tabular format with the columns containing the question, user answer, expected answer and the accuracy of performance of the model using this summary the user can self-evaluate his performance in our mock AI based interview.

As depicted in the Figure 1, the website provides smooth and a reliable user experience. Once the user clicks on the "Take Test" button, the interface navigates to a new route,

* Corresponding author: srilatha1727@grietcollege.com

© The Authors, published by EDP Sciences. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).
indicating the start of the test-taking process. At this point, the platform starts asking questions, and the user responds verbally. The user's responses are recorded by the AI and converted into text format using a speech-to-text conversion mechanism.

After recording the user's responses, the AI model evaluates them based on predefined metrics or criteria to determine their accuracy. The evaluation process may involve comparing the user's responses to a set of correct answers or a standard performance benchmark. Once the evaluation is complete, the AI model generates an accuracy score and displays the results on a new page. This process enables users to take tests or quizzes online and get immediate feedback on their performance. By providing a reliable and seamless user experience, the website aims to enhance user engagement and satisfaction, and promote learning and knowledge retention.

The machine learning model converts words into dense vector representations using an embedding layer. The embedding layer is essentially a matrix of weights that the model learns during training, and it maps each word in the vocabulary to a corresponding vector of a fixed size. The embedding layer performs the task of "embedding" each word into a continuous vector space. This means that it maps words into vectors in such a way that similar words are mapped to similar vectors. This is important because it allows the model to understand the meaning and context of words, and how they relate to each other in a sentence or a text document.

The vectors that are generated by the embedding layer are then passed through an encoder architecture, which consists of two sub-layers: the self-attention layer and the multi-head layer. The self-attention layer is responsible for learning the relationships between different words in a sentence or a text document. It does this by assigning weights to different words in the sentence based on their relevance to the overall meaning of the sentence. The multi-head layer then takes these weighted vectors and applies a transformation to them to create a final output vector that represents the entire sentence or text document.

2 Literature survey

2.1 Summary of existing work

The foundation for the transformer design, which is built utilising the Encoder and Decoder stacks as well as the Attention function, was developed by Ashish Vaswani and his colleagues Google researchers [1].

The goal is to employ a significantly quicker technique. Sequence modelling uses recurrent neural networks, extended short-term memory, and gated recurrent neural networks. The WMT English-German dataset, which contains around 4.5 million phrase pairs, and the English-French dataset, which has 36 million sentences, were both employed by the team. Tokens were broken into a 32,000 word-piece vocabulary. Transformer is a promising alternative for translation jobs since it trained more quickly than models based on recurrent or convolutional layers. All earlier models were outperformed by this one.

The Universal Transformer was created by Mostafa and his team members as an advance over the conventional transformers [2]. Utilising the parallel-in-time recurrent self-attentive sequence model and the idea of dynamic halting resulted in the biggest improvement. The group used the BABI question-answering dataset, which is made up of 20 distinct artificial tasks including stories, true-false questions, counting, etc. This research measures the subject-verb agreement in a sentence. The created model did well in the lambada language modelling challenge where it was necessary to determine the final word of the sentence.

The Bidirectional Encoder Representations from Transformers (BERT), a Google-developed open-source research, was created by Jacob Devlin and colleagues [3]. The basic
The goal of the BERT model is to comprehend the language so that many additional language activities may be carried out. The model is a group of encoder-based transformers that may be adjusted to carry out a variety of linguistic tasks. The Winograd NLI dataset (WNLI), Situations with Adversarial Generations dataset (SWAG), and Stanford Question Answering Dataset (SQuAD v1.1) are used to develop the model. This model is extensively used in many NLP applications, and Google uses it to improve search.

To examine the semantic similarity of the sentences, Reimers and Iryna [4] improved the BERT model and created a new model they called the Sentence BERT model. The goal is to confine the BERT model's computation to finding sentence similarity and to minimise the amount of calculations it does. The researchers reduced the amount of time the model needed to compute the answer by using the multi-Genre NLI datasets and the Stanford Natural Language Inference datasets. Performance has significantly improved from 65 hours to around 5 sec. The regression goal function of the model employs cosine similarity scores, while the classification objective function uses the SoftMax classifier.

Yu Zhang and his colleagues in the Google research team [5] developed a semi-supervised learning-based Automatic Speech Recognition model (ASR). The objective was to improve the wav2vec 2.0 model's performance in speech recognition analysis. A conformer encoder and an LSTM decoder architecture were used. The phases in the approach include space augmentation, batch-wise mixing, language modelling and fusion, pre-training parameters, and fine-tuning parameters. The method's final outcomes include space augmentation, batch-wise mixing, language modelling and fusion, pre-training parameters, and fine-tuning parameters. Authors [6] highlighted the significance of ML in prediction, pattern recognition and error reduction across diverse fields, emphasizing the impact of AI in broad domain. Authors [7] suggested web-based chatbot that employs Machine Learning, particularly Random Forest, to recognize and address user emotions based on text and image inputs. It offers personalized solutions for managing emotions effectively.

### 3 Problem statement and objectives

The objective is to help students and professionals prepare for interviews, as they can be stressful and challenging for most people. The web-based platform aims to provide users with an environment that mimics the interview experience. The machine learning model, integrated into the platform, generates a series of questions that simulate an actual interview. The use of this model allows users to practice their interview skills and receive feedback on their responses, which can help them identify areas for improvement. The platform is designed to be user-friendly and accessible, ensuring that users feel comfortable and confident while using it. Objectives are as follows:

- To familiarise users with the interview process, boosting their confidence and increasing their chances of success.
- To offer users a streamlined and user-friendly experience that allows them to confidently and easily prepare for interviews.
- To remote accessibility for users by allowing those to take the interview at their own pace and in their own preferred location. This eliminates the need for candidates to travel long distances or take time off work to attend an in-person interview.

### 4 Proposed method

Figure 2 shows the front-end to provide a user interface (generally a website) through which a user can take his/her test and can evaluate themselves with the accuracy. In addition to providing text-to-speech functionality, the front-end of the application also includes a user
input section where the user can provide their answers to the questions. To facilitate this, we used a form element with various input types such as text boxes, radio buttons, and checkboxes depending on the type of question being asked. To provide immediate feedback to the user, we used JavaScript to validate their responses and display a message indicating whether the answer was correct or incorrect. Additionally, we implemented a timer feature to ensure that users are aware of the time they have left to complete the test, and a progress bar to show how much of the test they have completed.

Once the user responds to the question, the spoken response is captured using the Web Kit Speech Recognition module. The captured response is then compared to the correct answer stored in the database using the SBERT algorithm. Based on the similarity score, the response is marked as correct or incorrect. The user is then provided with feedback on their performance, including their score and the correct answer to each question. This feedback allows users to identify areas where they need improvement and to focus on those areas in future practice sessions.

The website also provides users with the option to review their performance history and track their progress over time. This feature allows users to see their improvement over time and to identify areas where they may still need additional practice. Here, we use machine learning algorithms to evaluate the user's responses and generate a score. The score is then sent back to the front-end, where it is displayed to the user along with a summary of their performance.

To ensure that the user's data is secure, we have implemented various security features such as encryption of data during transmission and storage, and access control measures to restrict unauthorized access to the data. Overall, the web-based platform provides users with a comprehensive and accessible tool for preparing for interviews. By integrating machine learning, NLP, speech recognition, and speech synthesis technologies, the platform offers a cost-effective and time-saving alternative to traditional interview preparation methods. The user-friendly interface, automated interview process, and personalized feedback and progress

Fig. 1. Homepage of the proposed work.

Fig. 2. Architecture diagram.
tracking features make it an ideal tool for students and working professionals seeking to improve their interview skills and boost their confidence.

4.1 Responsive voice API

The Responsive Voice API is a JavaScript library that provides an easy-to-use API for adding text-to-speech functionality to web applications. The library can be used to generate spoken output from text strings using a wide range of languages and voices. In this way, developers can provide more accessible and interactive experiences for users. The library works by integrating with the Web Speech API, which is a set of JavaScript APIs that enable speech recognition and synthesis in web browsers.

4.2 Web speech kit

Web voice API is a JavaScript API that allows web developers to integrate voice detection and synthesis into their websites. Modern web browsers such as Google Chrome, Mozilla Firefox, Microsoft Edge, and Safari support it. The Web Speech API has been split into two components: Speech Recognition API and Speech Synthesis API.

4.3 Speech recognition API

The Speech recognition API allows web applications to recognize speech input from users. It provides access to the device's microphone, which captures the user's spoken words and converts them into text. The API then returns the text as a string that can be used by the application. The Speech Recognition API can also recognize different languages and dialects. The supported languages depend on the browser and the operating system. Using the Speech Recognition API, web developers can create voice-controlled applications that can be used without the need for a keyboard or mouse.

4.4 Speech synthesis API

The Speech Synthesis API allows web applications to convert text into speech. It provides access to the device's speakers, which play back the synthesized speech. The API can be used to generate speech in different languages and voices. The supported voices depend on the browser and the operating system. Using the Speech Synthesis API, web developers can create applications that can speak out text to the user. This can be useful for users with visual impairments or for applications where hands-free operation is required.

4.5 Django backend

Django, a popular Python-based web framework, is used to quickly and efficiently construct online applications. According to the Model-View-Controller (MVC) architectural pattern, the programme is separated into three major parts: the model (data), the view (user interface), and the controller (logic). Django is used in the backend to manage the application's server-side logic in the context of the automated interview assessment research.

4.6 SBERT model

An example of a natural language processing job that uses the SBERT (Sentence-BERT) model includes sentence categorization, question answering, and semantic textual similarity.
The BERT (Bidirectional Encoder Representations from Transformers) model, which is frequently employed for a variety of NLP applications, is extended in this. Similar to BERT, SBERT is a pre-trained deep neural network model, which means it has already learned how to encrypt the meaning of language into a collection of numerical vectors by training on a vast quantity of text data. SBERT is especially made for encoding sentence-level embedding, which are representations of full sentences in a lower-dimensional space that capture the semantic meaning of the phrase. This is a significant distinction between SBERT and BERT.

5 Results and discussions

5.1 Description about dataset

In the present research work, we are using MongoDB database for storing question answer pairs. The questions related to a topic such as DBMS are stored in a collection (table). To populate the database with relevant and high-quality questions, we scrape interview question websites that contain questions from previous years. This ensures that our database contains questions that are relevant to the user and are of good quality.

Fig. 3. Representation of database.

As represented in Figure 3, the database contains two attributes a question attribute and an answer attribute. We can increase or decrease the number of questions as per the requirement we can add collections and remove collections as per the requirement as the database is placed in cloud we can dynamically add and delete values without any errors. MongoDB allows us to store and retrieve data quickly and efficiently. It is a scalable and flexible database that is suitable for handling large amounts of data. By using MongoDB we can ensure that the user is presented with a variety of questions on different topics to help them prepare for their interviews. The accuracy of the model performed is given by the Spearman's correlation of 0.904 which is the highest among all the available machine learning models. In the figure 5 our AI is taking an interview to the user and is collecting the response of the user’s voice input after the user clicks on the answered button the user’s input is sent back to the user and then is sent to the backend server for evaluation.

5.2 Significance of the proposed work

Sentence-level embeddings are created using the SBERT (Sentence-BERT) approach, which employs BERT (Bidirectional Encoder Representations from Transformers). The importance of this approach lies in its capacity to compare sentences on the basis of their meanings rather
than merely their syntax or structure. Applications like text categorization, information retrieval, and question-answering can all benefit from this.

Fig. 4. Live interview.

Fig. 5. Results of our Interview.

5.2.1 Responsive voice API

The Responsive Voice API is a text-to-speech API that allows for the generation of natural-sounding voices for use in applications. This method is significant because it enables the creation of more engaging and interactive interfaces, particularly for users who may have difficulty reading or understanding written text. In the context of the Interview Automation System, the Responsive Voice API can be used to deliver questions to users in an audio format, providing a more realistic interview experience.

5.2.2 Web speech API

The Web Speech API is a browser-based API that enables the recognition of speech input and the generation of speech output. This method is significant because it allows for the integration of speech recognition and synthesis capabilities into web applications, without the need for additional software or plugins. In the context of the Interview Automation System, the Web Speech API can be used to collect user responses in an audio format, enabling more accurate evaluation of interview performance.

5.2.3 Django backend

Django is a web framework for Python that provides a set of tools and libraries for building web applications. This method is significant because it allows for the development of scalable and maintainable web applications, with built-in support for common web development tasks such as database management and user authentication. In the context of the Interview Automation System, the Django backend can be used to manage the storage and retrieval of user data, and to provide a secure and reliable platform for the delivery of interview content.
6 Conclusion and future enhancements

The Interview Automation System is an innovative and effective tool for users to improve their interview skills. Our platform utilizes AI technology to provide a realistic interview experience, allowing users to practice and refine their responses. Through our data pre-processing and database management, we ensure the accuracy and relevance of our questions and answers. Our audio-based question delivery and collection method provides a seamless and user-friendly experience. Our platform’s scalability and potential for additional functionalities make it a versatile tool for individuals and organizations alike. Ultimately, the research aims to empower users to feel confident and prepared for real-life interviews, leading to increased success in their career pursuits. In conclusion, the Interview Automation System provides a valuable tool for users to prepare for real-life interviews. The research workflow involves several key steps, such as data pre-processing, audio-based question delivery, and answer collection and evaluation. The system is easily scalable and has the potential to include additional functionalities such as users’ confidence detection, resume-based topic questioning, and body language analysis. Scaling the research can be done by adding different domains related topics, which will allow users to practice questions specific to their field of interest or expertise.

References

5. Y. Zhang, J. Qin, D. S. Park, W. Han, C. C. Chiu, R. Pang, Q. V. Le, Y. Wu, Arxiv, 10504, 10 (2020)