Title	**Interview Chatbot Proposal**
Course	CS/QTM/LING-329: Computational Linguistics
Project	https://github.com/raufai02/CS329
Authors	Rauf Iftikhar, BA in Computer Science, BA in Philosophy, `riftikh@emory.edu`
	Ameer Husary, BS in Computer Science, minor in Data Science, `ahusary@emory.edu`
	Noah Okada, BS in Neuroscience, BA in Computer Science, `nokada@emory.edu`
	Bilal Zaidi, BS in Computer Science, `mhaide8@emory.edu`

Abstract

Here we introduce a novel dialogue system designed to perform behavioral interview training for job-seekers. Our system, InterviewBuddy, is designed for upperclassmen undergraduates and recent graduates who are entering the software engineering job market. Such individuals are less experienced and thus more prone to face difficulty in the interview process. Our tool is designed to help this demographic practice responding to questions and receive constructive feedback to help them improve their interview skills, and thereby their chances at landing a job. Our dialogue system mimics an interviewer and simulates a job interview with the user, followed by an evaluation of the user's candidacy by the system. The dialogue system is implemented using emora_stdm framework and the OpenAI API, as well as custom ontologies and macros. Ontologies help personalize the interview experience by matching specific user-generated traits, facilitating generation of a simulated interviewer persona. Macros that call the OpenAI API are implemented to select interview questions from a pre-defined bank based on user responses and generated quantitative measurements of user performance. The dialogue system is evaluated both by users, providing a valuation of user satisfaction, as well as the objective quality of the system. Our novel tool is the first interview dialogue system that is candidate-oriented, and thus has the potential to create a new market for job seekers looking for a competitive advantage.

1 Team Vision

1.1 Concept

Job interviews can be a daunting and stressful experience, especially for those who are just entering the job market or lack experience in a particular field. Additionally, the competition for jobs is high: on average, a corporate job listing in the United States receives 250 applicants, with each position interviewing only 2 to 6 candidates[4]. In such a competitive environment, it is crucial for candidates to be well-prepared and make a positive impression during their job interviews. However, many students and job seekers receive inconsistent or inadequate interview training, leaving them ill-equipped to navigate the difficult interview process. This lack of preparation can cause anxiety and frustration, and can result in missed job opportunities.

To address this challenge, there is a growing need for innovative solutions that can provide job seekers with the tools and resources they need to prepare effectively for interviews. We developed InterviewBuddy, an AI-powered tool designed to assist novice interviewees in practicing and improving their ability to respond to interview-style questions, with the aim to better prepare them for job interviews. By providing concrete feedback on a user's performance, InterviewBuddy will help users develop the skills and confidence necessary to excel in any interview setting.

We aim to create a user-friendly and interactive chatbot that can engage with interviewees in a personalized and effective manner. By pairing advancements in NLP tools with research from social psychology and organizational management, InterviewBuddy will be able to intelligently provide targeted feedback to improve interview and communication efficacy.

A session with InterviewBuddy will be broken into three phases: introduction, interview, and feedback. In the pivotal interview phase, users will interact with the dialogue agent by typing their responses to interview questions that are tailored to measuring their cognitive, technical, leadership, and interpersonal (culture fit)

abilities. This will give users a chance to familiarize themselves with the types of things they want to talk about during an interview. Not only will the user have the chance to practice interview questions on the spot, but they will also receive personalized feedback via our user feedback feature. This feature provides the user, upon completion of the interview, an series of scores that reflect how their performance in key areas that have been studied in literature as significantly affecting hiring likelihood. Providing users with dynamic feedback after every interview session makes InterviewBuddy extremely valuable for applicants who would like to know exactly what they can improve on in the future. Through InterviewBuddy we hope to provide science-based interview training that will allow all applicants to enter their interview with a sense of reassurance knowing that they are now prepared to conquer any job interview.

1.2 Target Audience

InterviewBuddy, in the long-term, has the potential to be scaled to serve anyone who is interested in developing their interview skills and can benefit a variety of people ranging from recent college graduates to anyone interested in returning to the workforce or switching jobs. Having this user flexibility requires massive amounts of data and programming. Therefore, for the current project, we focused on a limited scope tool specifically for undergraduates and recent graduates entering the software engineering marketplace.

We selected our target audience carefully to create a chatbot that can benefit others by leveraging our team's collective experience with the software engineering interview process. As per DataUSA, the software development industry in the US had approximately 1.42 million developers in 2020, with a significant gender and ethnic imbalance. About 81.9% of the developers were male and 18.1% were female, while 53.4% were White (Non-Hispanic), 33.6% were Asian (Non-Hispanic), and 4.06% were Black, among others [2]. Taking this into account we believe that our open-source free to use chatbot is necessary, as it breaks the cost-barrier of entering the software engineering world. InterviewBuddy could be used by applicants of different backgrounds and help them succeed in their software engineering interviews by developing their interview skills and providing a simulated software engineering interview; thereby creating a more diverse and inclusive working environment in the software engineering world. According to Goremotely, there were approximately 1,365,500 job openings for software developers, and the industry is expected to grow by 21% . In 2017, there were only 50,000 Computer Science graduates, but over 500,000 job openings, highlighting the potential for growth in this sector [1]. Given these insights, our chatbot has been developed to cater specifically to the needs of this high-growth potential audience.

2 Challenges

2.1 OpenAI API

During the process of implementing the GPT3 API for natural language understanding, our team encountered several challenges. The first obstacle we faced was reducing token usage to speed up API calls and minimize costs. To address this, we made modifications such as maximizing tasks on each input prompt and indexing our transcripts to match the output locally. By doing so, we were able to significantly reduce the number of API calls needed and improve the efficiency of each call by returning only the index of user responses from the interview transcript. In fact, we were able to reduce the number of API calls by three-fold, as we only had to call a single method to evaluate the entire transcript.

To further increase temporal efficiency, we implemented intelligent model choice by analyzing each function that used the GPT3 API for task complexity. We determined which functions were simple enough to utilize less robust models such as TEXT-DAVINCI-003, while we reserved the more powerful TURBO-3.5 model for larger tasks. This strategy allowed us to significantly decrease the delays experienced by the user during the interview process.

Despite our modifications, we still encountered challenges such as RateLimitErrors from the OpenAI API, which slowed down the debugging process. Unfortunately, there was no way to catch these errors inside an

EXCEPT block in Python. Despite these setbacks, our team remained committed and ultimately succeeded in implementing the GPT3 API for natural language understanding.

3 Dialogue Overview

Our dialogue system is designed to guide users through a triphasic process, consisting of three distinct phases. The first phase is the Introductory Phase, where the system introduces itself and explains the purpose of the conversation to the user. This phase aims to establish a comfortable and welcoming environment for the user to engage in the dialogue. The second phase is the Interview Phase, where the system begins to ask questions to gather information from the user. This phase is the core of the conversation, and the system uses its natural language processing capabilities to understand the user's responses and generate follow-up questions that aim to extract the necessary information. Finally, the third phase is the Feedback Phase, where the system provides the user with a summary of the conversation and asks for feedback. In this phase, the system can refer to specific examples from the interview phase to provide context for its summary. This phase aims to ensure that the user has provided all the relevant information and to give them the opportunity to provide feedback on the conversation.

Overall, the triphasic design of our dialogue system is intended to provide a structured and effective way for users to communicate with the system and achieve their desired outcomes. By breaking down the conversation into distinct phases, the system can guide the user through each stage, ensuring that all necessary information is gathered, and providing a summary to ensure that the user's needs have been met.

3.1 Introductory Phase

During the introduction phase, our system will greet itself to the user and gather information about the user: user's name, background, and job interests. In designing the Introduction Phase of our dialogue system, we aimed to achieve two primary objectives: firstly, to provide the user with a clear understanding of the purpose and capabilities of InterviewBuddy, and secondly, to establish a sense of emotional connection with the user and provide encouragement before the simulated interview.

The system begins by introducing itself to the user as InterviewBuddy and describing how it will help the user. The system then asking the user about their name, mood, major and career interests. Before moving to the interview, the system asks the user how they are feeling before the beginning. If they feel ready or otherwise, the chat bot will acknowledge them and start the interview. However, if the user says they are feeling anxious or worried about the interview, InterviewBuddy will acknowledge that and will generate words of encouragement to ease the user's anxiety as they are about to begin the interview. After this, the interview will start and the user will be introduced to their interviewer.

Sample Dialogue

S: Hello, I am InterviewBuddy. I am an interview chatbot that is designed to help interviewees practice their interview skills in order to better prepare for the real thing. You're here for the interview today, right? What should I call you?
U: Call me Bilal
S: Nice to meet you, Bilal! How are you feeling right now?
U: I am feeling very good right now! Thank you for asking!
S: That's awesome! I'm glad you're feeling well! You're a young college student/graduate, correct?
U: Yes, I am. I am currently a senior at Emory University
S: Gotcha. What is your major, if you don't mind me asking?
U: I am a Computer Science major planning to get my bachelors of science by Fall 2023!
S: Perfect! computer science is a great area of study for a position in software. What kind of software developer do you want to be when you start applying for jobs?
U: I've always been interesting in being a SQL developer. I've always liked working with data.
S: And what field of software engineering/computer science are you interested in?
U: I would say data science as that aligns with the kind of software developer I want to become in the

future
S: So you want to get into data science , huh? That's awesome. Always found that line of work to be pretty interesting. Anyways, we should probably get into the interview. Are you feeling nervous or confident?
U: Ehhhh, I'm feeling a bit anxious to be honest. Not really sure what to expect and if I'm up for it
S: Don't worry about being perfect. Just be yourself and let your personality shine through. That's what will make you stand out. Are you ready now?
U: Thank you for the kind words! I think after hearing that, I feel a bit more confident!
S: Thanks for sharing. Now we're going to begin
U: Ok then!

3.2 Interview Phase

The second phase is the interview phase. During the interview phase, our system asks the user a series of directed questions to assess the user's responses. The questions are divided into four main categories:

- Technical/Role Specific
- Leadership/Independence
- General Cognitive Ability
- Culture Fit

Based on the information in the user's responses to the introductory questions, our system will intelligently select an interviewer persona that matches the user's interests in a specific computer science field. This persona will have a name, job, and set of skills that are used to relate to the interviewee. The persona will greet the user and make the conversation seem more natural. Having this feature will allow us to ease the user into the actual interview process, create a personalized user experience, and enhance the contextual awareness of our bot. The persona will start by asking the user a question that falls into one of the categories listed above. As the user answers, our bot will also intelligently determine which follow-up question best matches the current context that the user is describing. Moreover, our bot will intelligently determine which comment or response it should give back as an affirmation to the user, by analyzing the user's answers, before asking the next question and creating a natural back-and-forth. An example would be, "That's a very interesting project! Reminds of some work I did back in college" or "I remember when I had to code in ($language)". We repeat this process and build conversation! Having these features will help maintain that natural sense of conversation. The dialogue system will attempt to cover all four main categories. We can see the process in the sample dialogue below and it repeats to create the conversational feel of an interview as it covers the main categories.

Sample Dialogue

S: Hi there! My name is Charlie, I know a thing or two about artificial intelligence/machine learning. I am working at Mathworks as a Design Group Engineer. I will be conducting the interview with you! I want to you to know that I am on your side throughout this process, just do your best when answering the questions. So let's start!
U: ok!
S: Tell me about a project you are currently working on
U: I am currently working on a database project using firebase and SQL to analyze the customers of a company.
S: Why did you choose to use the programming language you're currently using for this project?
U: I chose SQL because it is the most efficient when it comes to analyzing tables.
S: I remember when I first learned to program in that language. It's a hard language to learn I'm impressed that you used it! Can you tell me about a time when you had to work independently to complete a task?
U: ...

3.3 User-Feedback Phase

The third phase we will be implementing is our user-feedback phase. During this phase, the dialogue system will be quantifying the quality of the user's responses and give them concrete feedback as to how they should respond differently. As the user finishes their interview with our system, the chatbot will provide the user several interview scores reflecting the quality of the different dimensions of their interview. These scores will be determined based on numerous factors, such as the length of the user's response, how quickly the user responds to questions, the sentiment and the way the user responded, and the relevance of the user's answer to a question. Using this process to determine the quality of the user's interview skills will help interviewees to see how they can improve for a non-simulated interview. This will also help those who are new to being interviewed and will give them some idea of what they need to prepare for and improve on. A new score is generated every time the user decides to use the chatbot for a simulated interview. In other words, our chatbot will not be storing any user data. We chose to do this to remove any bias from data that could have been stored, and to maintain transparency and consistency for users that would like to practice an interview again.

Sample Dialogue

> S: Congratulations on completing the interview! I can now analyze your responses to give you some feedback on points to improve. Would you like to receive feedback?
> U: yea
> S: What area would you like feedback on? [job requirements, friendliness, emotional content, inclusivity, efficiency, unique words]
> U: Can we talk about friendliness?
> S: Studies in psychology have shown that interviewers are more receptive towards friendly candidates. Here we analyze the conversation to see how friendly you were during the interview. Your overall friendliness score was: 0.6 out of 1.0 possible. Would you like to see some specific examples?
> U: yes
> S: Here is an example of a response where you demonstrated friendliness:11 U: working in a remote environment is efficient because it reduces commute times and allows me more flexibility with my schedule however I like the accessibility of coworkers in an in person environment.
>
> In this example you received a score of: 1.0. Do you want to see another example?
> U: Yes!

4 Methodology

4.1 Introduction Phase

The introduction phase was designed to keep the user engaged in the conversation with our chat bot. It would gather basic information about the user that we would store and then later use that data to use a personalized interviewer that shares a similar persona and interests to the user when it comes to computer science. The beginning of the introduction phase gathers the user's name by using the macro MacroGPTJSON. This macro puts a prompt into ChatGPT that asks how the user wants to be called and then returns that output. The output from GPT is what we use to gather what our system should call the user. We store that name in a pickle file, so that if the same user chooses to return to use our chat bot again, it will greet that user slightly differently to show that it is aware that this is a returning user. We store the user's name by using two functions, saveName() and loadName(). The saveName() function saves the variable that stores the user's name in a pickle and loadName() checks that pickle to see if that name has been stored in there. When the pickle file doesn't exist, there is a condition that will use saveName() to create that pickle file. Then, in all other cases, loadName() is called at the end of the chat bot to save the user's name and other information about the user. Then after InterviewBuddy has fully ran, we use the saveName() function to save all the information that the user inputted in the dialogue flow, including the name of the user. We also use MacroNLG() to output the user's name in the introduction phase. The dialogue flow in the introduction

phase also checks for fields of computer science and job types that the user inputs when asked about them in our chatbot.

We created a file `major_ontology.json` that has an extensive ontology of many different fields in computer science and different types of jobs one could have with a programming background. Then in our introduction phase, we match the user's input to one of those fields, and later use that information to match the user with an interviewer persona that has similar interests to the user. Lastly, the introduction phase also uses a macro to make InterviewBuddy appear more personal and human-like to the user. This macro is called MacroEncourage(). When the chat bot asks how the user is feeling before the interview, if the user responds that they are feeling anxious or nervous, this macro is called and generates words of encouragement to ease the user into the interview and make them feel more comfortable before the beginning of the interview process.

4.2 Interview Phase

After the introduction phase we have different data structures and macros that guide the user through an interview phase. We start by making a data structure that involves a dictionary of dictionaries, in the question_bank.json file, where the first keys are the four categories of questions that are a part of the interview phase (technical, cognitive, culture-fit, and independence/leadership), and the values are dictionaries where the second keys are the big generic interview questions, that relate to each category, and the values are lists of follow-up interview questions that are a bit more detailed and ask more information about the user's experience.

MacroGetBigQuestion() loads the question_bank.json and selects one of the big four categories, and then removes the key questions so it does not appear again. The macro then returns one of the big questions in the transition and stores the follow-up questions of the key in the global vars to be accessed later. MacroGetLittleQuestion() is a function that pulls the list of follow-up questions from vars. In this function, we intelligently select one follow-up question via GPT3 as a context analyzer to select the best match from the list, and return the follow-up question in the transition.

We also created a file `personas.json` that contains a dictionary data structure that consists of different keys of the different specializations of computer science (e.g. AI/ML, database). Specifically, we created 10 interview personas in persona.json that were assembled from the 16 different job descriptions gathered from profiles and job listings on LinkedIn. The values are the different fields of the persona that relates to each tech field. Fields such as the name of the persona, skills they have, company and position they are in, etc. We used these descriptions and requirements to personalize the interview experience to the user using the macro MacroPersona().

MacroPersona() takes in the user's responses to the fields they are interested in as the context, and then passes it to the OpenAI API alongside a list of the keys (computer science fields) from personas.json. We utilize the GPT3 model to return the index of the field that best matches the user's interests which we can then use to provide the user with an interviewer persona that matches their interests. We also wanted to enhance the conversational component of our bot. This is where the file contextual_comments.json helps as it contains about 30 responses that are generic enough to fit the user's answer to a follow-up question. MacroRespond(), similar to MacroPersona(), uses GPT3 to determine the best system response to the user based on their answer to a follow-up question. These macros allow us to personalize the experience of the user and create contextual awareness.

4.3 Feedback Phase

To assess an applicant's performance on an interview, we analyzed their performance across three categories: response quality, affective expression, and the candidate's fit towards the listed job. These categories were selected by examining the four-stage employment interview model proposed by Wiersma, which highlights the different direct and indirect attributes that are assessed throughout the stages of an interview [8]. Furthermore, to ensure that our evaluation methods adequately reflected tools used in previous research we employed insights from Naim et al.'s toolkit for automated analysis of job interview prediction to [5]. Drawing from

these insights we engineered a set of prompts to direct the GPT-3.5-Turbo model to analyze the interview transcript based on the performance categories.

To evaluate response quality we designed our prompts to analyze the transcript for lexical density, the number of unique words, and the inclusivity of the language used by the user. These features were designed to assess a job candidate's perceived intelligence by measuring the efficiency and variation of words in their speech [7]. In addition, we evaluated response inclusivity based on results from Naim et al. which showed that individuals who utilized more inclusive language had a significantly higher probability of being hired than those who used individualistic language in their interview responses [5]. The lexical density, proportion of unique words, and proportion of inclusive language were stored as score values (floats) to be returned to the user during the evaluation dialogue.

We chose to evaluate affective expression, because previous research has shown that interview success (resulting in subsequent job offers) is positively correlated with the perceived friendliness and positivity of a candidate [6, 5, 3]. To assess a user's positive sentiment in the interview we designed our prompts to evaluate each response for the ratio of positive emotional words (e.g. hope, improve, kind) to negative emotional words (e.g. bad, fool, hate) used throughout the interview transcript. These proportions were stored as score values (floats) with particularly good and bad responses stored as separate examples for the user.

To assess a candidate's fit for the job we designed prompts to compare the interview transcript with the job description for the interviewed position. We analyzed each individual response for the number of elements in the job description (i.e. technical abilities, preferred experience) that the responses mentioned. Furthermore, we compared the total transcript to the job description to see the total number of elements fulfilled by the conversation during the interview. We then stored these scores and examples of positive (responses that listed items from job requirements) and negative (responses that did not mention any job requirements) responses as examples for the user. We developed the MacroGPTEval as a Macro to score the users responses and store information about this score to provide to users. This Macro was designed to send the entire transcript of the interview phase to be evaluated based on the prompts (see Fig 2). For each category, we asked GPT3 to give quantitative scores and the indexes of specific examples that illustrate strength and weakness in the area evaluated. Moreover, in some cases we also requested single adjectives to describe the response. In total we stored 27 variables from the evaluation Macro.

Job Fit

- Compare each user response with the job description. Rate each response based on the number of elements they fulfill from the job description. Only for the highest scored question and the lowest scored question, return the index of the response and provide a list of requirements they fulfilled.
- Compare the entire transcript with the job description. Rate each response based on the total number of elements they fulfill from the job description. Return a float value representing the number of items matched divided by the number of possible elements for the entire transcript.

Response Quality

- Evaluate the information density of each response. Calculate a float value rating for the efficiency of conveying information on a scale of 0 to 1 and provide the index where the answer occurred. Respond with the index of the most efficient statement and the index of the least efficient statement by the user. Also calculate the lexical density of the response
- Evaluate the total number of unique nouns, adjectives, and adverbs used in the transcript. Calculate a float value indicating the total number of unique words divided by the total number of words. Respond with the score and the most frequent word and least frequent word from the group of nouns, adjectives, and adverbs
- Evaluate the response for inclusive language. Count the total number of individualistic pronouns (e.g., 'I', 'me') compared to the number of inclusive pronouns (e.g., 'we', 'us'). Calculate a float value indicating a rating for inclusivity on a scale of 0 to 1. Respond with the index of the most and least inclusive answer and a list of adjectives to describe the quality of the response.

Affective Expression

- Please analyze the responses from this transcript for emotional content. Calculate a float value rating each user's response from 0 (negative emotional content, e.g., bad, fool, hate, lose) to 1 (positive emotional content, e.g., hope, improve, kind, love). Respond with the index of the the most positive emotional response and most negative emotional response and their scores. Also respond with a list of emotions expressed in the sentence.
- Please analyze the responses from this transcript for friendliness. Return a value for the overall friendliness of the text from 0(not friendly) to 1 (friendly). Respond with an example of the most positive friendly sentence. Also, respond with an example of the least friendly sentence. Return the index and a score of the friendly responses.

Figure 1: Evaluation Prompts

5 User Evaluation

Our product was designed with commercial potential as a product to be used by job seekers to boost their likelihood of employment. As such, interactive feedback regarding the quality of the dialogue system was essential for evaluation. Due to the multifaceted functionality of InterviewBuddy we chose to evaluate the chatbot based on several categories to assess its efficacy at the various stages of the interview and feedback. To ensure that the chatbot effectively communicated information during the interview we sought to understand the interpretability, naturalness, and informative nature of the conversation. Furthermore, to ensure that the experience was tailored to the individual needs of users we evaluated the chatbot's personalization, empathy, and efficacy at providing users with feedback. This evaluation was conducted by user testing the product in a cohort of users from our target audience (students seeking jobs in software engineering).

Our study cohort consisted of 40 students from Emory University. These students came from a variety of majors (neuroscience [n=2], psychology [n=1], computer science [n=24], quantitative sciences [n=13]), and were interested in pursuing a career in several different positions in software engineering (SWE) (front-end developers [n=5], data science [n=13], cyber-security [n=2], web developer [n=10], back-end developer [n=5], app/game developer [n=5]). The students completed 1 session (approx 15 minutes) with InterviewBuddy and then responded to a google form designed to assess the chatbots functionality on the categories mentioned above. To account for variability in subjective interpretation of the metrics we randomly deployed four different evaluation forms across equally distributed groups of participants (each group with 10 users). We then combined these results into a single metric that measured the user evaluation on a 10-point likert scale.

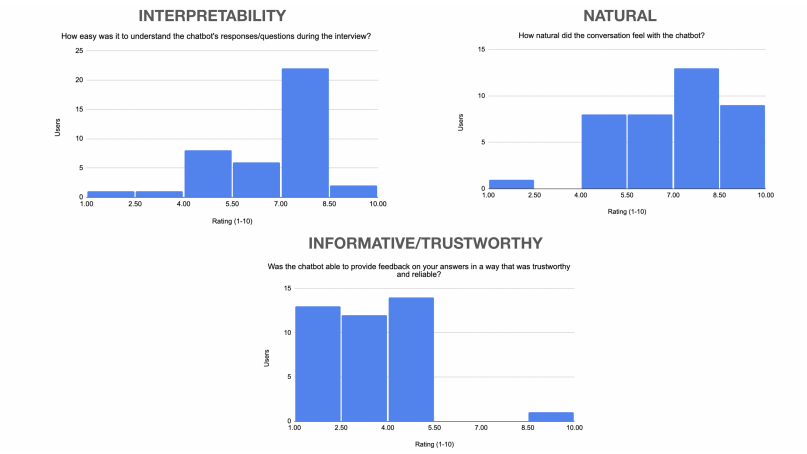

Figure 2: User Test Data (Interpretability, Naturalness, Informative)

The results indicated that during the interview the users felt that the bot was adequately interpretable and natural in conversation. In the context of InterviewBuddy interpretability was defined as the bot's efficacy in communication measured by correct syntax and grammar. Naturalness was defined as the feeling that the conversation was appropriate for a SWE interview and that the evaluation felt like a supportive human evaluator. The majority of users (n=32) indicated that the responses/questions provided by the chatbot were easily understandable, rating the interpretability greater than 6 on a likert scale of 10. Furthermore, many users (n=23) rated that the responses had a natural conversational tone (greater than 7) during the interview and evaluation. However, many users (n=27) indicated that the chatbot struggled to convey information in a manner that was "reliable and trustworthy" indicating that the chatbot was not effectively informative and trustworthy during the interview and evaluation. In the context of InterviewBuddy the informativeness of the chatbot was defined as the degree to which the responses and evaluation provided novel and insightful information about a user's interview performance. These results suggest that the interview phase of InterviewBuddy provided an effective simulation of a real-world interview, however the evaluation phase failed to provide sufficiently novel information for users to act upon.

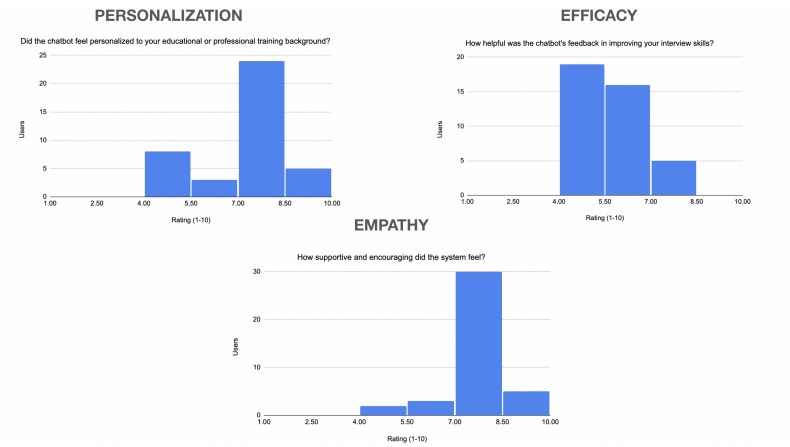

Figure 3: User Test Data (Personalization, Effication, Empathy)

The evaluation also showed that InterviewBuddy was effective at providing a personalized experience for users. Most users (n=29) rated the bot very highly (greater than 6) for its personalization to their professional background and job aspirations. Many users (n=23) rated the efficacy of the feedback for providing individually targeted suggestions highly (greater than 5). In the context of InterviewBuddy these results showcase that the interview and feedback phase were well suited to the responses of the user. Furthermore, the majority of users (n=36) rated the system as highly encouraging and supportive, indicating that the chatbot possessed a high degree of empathy. Together these results indicate that the dialogue system was effectively attuned to individual users and could provide a personalized interview experience.

6 Novelty

6.1 Competitors

The majority of interview simulator dialogue systems are designed by companies looking to hire candidates. In contrast, we are building a candidate-oriented dialogue system that helps interviewees improve their ability to answer questions. Comparable services that one can pay for are coaching services such as BetterUp that are restricted by time and money. Fortunately, our automated tool is both free to use and available 24/7.

The only existing automated tool that can be used to simulate an interview for a potential candidate is ChatGPT. ChatGPT can effectively simulate an interview but has a number of limitations. ChatGPT requires comprehensive prompt engineering, lacks built-in personalization, and is limited by the computational cost of a large-language model. Although we use GPT3 for parts of the interview and for generating data for the evaluation phase, we streamline the process of evaluation with automated prompt engineering, including generating the entire interview transcript as prompt context. Moreover, parts of our conversation use EMORA_STDM which runs significantly faster than GPT3. Another key features is that ChatGPT lacks a humanoid persona and speaks very unaffectedly. To make our chatbot anthropomorphic, we will give it the persona of an interviewer. This interviewer will have relevant experiences that can relate to the user and a specific role and position at a company that is closely related to the field that the user is pursuing. Overall, InterviewBuddy is faster and more personable than ChatGPT.

7 Contributions

Throughout the development of InterviewBuddy, every member of the team played a significant role in contributing to the success of the project. Each member brought unique skills and perspectives to the table,

and the synergy of their efforts culminated in a successful outcome.

Ameer made several significant contributions to the development of InterviewBuddy. Firstly, he wrote the emora_stdm dialogue transitions for the interview phase, which enabled smooth transitions between interview questions. He also wrote macros and created a question bank to be used for both the big and follow-up interview questions. Additionally, he created a persona json file that was used to give the interviewer a personality, and he wrote the contextual comments json file which was used to respond to the user with an affirmative response.

Noah was responsible for several key aspects of InterviewBuddy's development. He designed the system's overall architecture and prompt engineering for the evaluation phase. Additionally, he created the evaluation questions and wrote a macro that concatenated contextual comments. Noah's contributions ensured that the system was intuitive and user-friendly, and that the evaluation process was efficient and effective.

Bilal's contributions to InterviewBuddy included writing the MacroEncourage, which provided emotional support to the user during the introductory phase. He also wrote the emora_stdm transitions for the introductory phase, enabling a smooth transition to the interview phase. Furthermore, Bilal developed a function to recognize returning users, which ensured that users could resume their progress from where they left off in their previous session.

Rauf made a significant contribution to InterviewBuddy by designing the system's overall architecture. He compiled a database of job descriptions that was used in the interview phase, and he wrote a macro that called GPT to select a follow-up question based on context. Additionally, he converted existing functions to analogous Babel-bot functions and wrote transitions for the evaluation phase. Finally, he wrote the MacroWhatElse for the evaluation phase, which helped the system to respond to the user's follow-up questions. Rauf's contributions were essential in ensuring that InterviewBuddy was effective, efficient, and user-friendly.

Overall, the combined efforts of our team resulted in a sophisticated and effective natural language understanding system that helped users improve their interview skills. Each member's unique contributions were critical to the project's success, and the collaboration amongst the team was effectively designed so that the work was distributed equally.

Member	Action	Contribution
Noah	- Prompt Engineering (evaluation & question selection) - High-Level Design	25%
Bilal	- Introduction Transitions - Ontologies - User-Data Collection	25%
Ameer	- Interview Q's & Follow Up - Interview Dialogue Transitions - Ontologies	25%
Rauf	- Job description - Evaluation Design - Dialogue Flow Cohesion - Macros - Storing/Loading Data - Merging Parallel Work	25%

Figure 4: Member Contributions

References

[1] Djurovic Ana. New Intel - How Many Software Engineers Are There in 2023? 03-07-2023. URL https://goremotely.net/blog/how-many-software-engineers-are-there/. Accessed on 03-12-2023.

[2] DataUSA. Software Developers. n.d. URL https://datausa.io/profile/soc/software-developers#demographics/. Accessed on 03-12-2023.

[3] Patrick Gebhard, Tanja Schneeberger, Elisabeth André, Tobias Baur, Ionut Damian, Gregor Mehlmann, Cornelius König, and Markus Langer. Serious games for training social skills in job interviews. volume 11, pages 340–351, 2019. doi: 10.1109/TG.2018.2808525.

[4] Dalton M.R. and Groen J.A. How do jobseekers search for jobs? New data on applications, interviews, and job offers. 2020. URL https://www.bls.gov/opub/btn/volume-9/how-do-jobseekers-search-for-jobs.html. Accessed on 03-12-2023.

[5] Iftekhar Naim, Md. Iftekhar Tanveer, Daniel Gildea, and Mohammed Ehsan Hoque. Automated analysis and prediction of job interview performance. volume 9, pages 191–204, 2018. doi: 10.1109/TAFFC.2016.2614299.

[6] Tanja Schneeberger, Anke Hirsch, Cornelius König, and Patrick Gebhard. Impact of virtual environment design on the assessment of virtual agents. In *Proceedings of the 19th ACM International Conference on Intelligent Virtual Agents*, IVA '19, page 148–150, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450366724. doi: 10.1145/3308532.3329455. URL https://doi.org/10.1145/3308532.3329455.

[7] Raghu Verrap, Ehsanul Nirjhar, Ani Nenkova, and Theodora Chaspari. "am I answering my job interview questions right?": A NLP approach to predict degree of explanation in job interview responses. In *Proceedings of the Second Workshop on NLP for Positive Impact (NLP4PI)*, pages 122–129, Abu Dhabi, United Arab Emirates (Hybrid), December 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.nlp4pi-1.14.

[8] Uco-Jillert Wiersma. The Four Stages of the Job Interview: Helping Interviewers Put Two and Two Together. 2015. URL https://doi.org/10.1108/EBHRM-11-2015-0045. Accessed on 03-12-2023.