

Development of LLM-Based System for IT Talent Interview

David K. H. Pasaribu

School of Electrical Engineering and
Informatics

Institut Teknologi Bandung

Bandung, Indonesia

davidkarel.h.pasaribu@gmail.com

Agung Dewandaru

School of Electrical Engineering and
Informatics

Institut Teknologi Bandung

Bandung, Indonesia

dewandaru@itb.ac.id

G. A. Putri Saptawati

School of Electrical Engineering and
Informatics

Institut Teknologi Bandung

Bandung, Indonesia

putri@itb.ac.id

Abstract—The recruitment process usually requires a lot of resources and time. In the IT field, recruiters must also have domain knowledge in the IT field to be able to recruit well. Technological developments have given rise to various technologies that can help recruiters in the recruitment process. In the proposed approach, interviews are carried out by conducting interviews for each competency point given by the recruiter. Interviews consist of 2 stages, namely behavioral and technical. Behavioral interviews were conducted using BEI (Behavioral Event Interview) using the STAR (Situation, Task, Action, and Result) method. Behavioral and technical interview assessments are carried out using a 2-class classification. The results of the classification are aggregated to get a score for the whole interview. Few-shot learning was carried out on GPT-4o to generate interview questions. Evaluation with G-Eval on STAR question generation showed answerability metric score of 0.8938, listening metric score of 0.9633, relevance metric score of 0.986, and STAR completeness metric score of 0.9532. Interview evaluation or scoring was carried out by fine-tuning the Longformer model for behavioral interview assessment and few-shot learning on LLM GPT-4 for technical interview assessment. The use of RAG using GLiNER on a knowledge base containing knowledge about terms in the IT field is carried out to help augment the output of Longformer. The knowledge base used is Wikipedia. The data used in the fine-tuning process is synthetic data created with prompt engineering on GPT-4 and GPT-4o. The results of fine-tuning the Longformer model to classify behavioral interview results have an F1-score of 0.962, an accuracy value of 0.967, a precision value of 0.970, and a recall value of 0.967.

Keywords—interview, IT talent, Large Language Model, few-shot, GPT

I. INTRODUCTION

The current business and technology environment has faced significant challenges in accommodating the needs of professionals in the field of Information Technology (IT). Rapid growth in the IT sector, driven by technological innovation, digital transformation and industrial evolution, has resulted in an increase in demand for skilled and competent talent in various IT disciplines. Technology-based organizations, companies and start-ups are increasingly competing for qualified human resources, who have the necessary technical knowledge and can meet the complex demands of the digital world.

The increasing need for IT professionals and the need for the right talent acquisition has driven the need for automated assessment systems. In the current conditions, the IT industry is facing a shortage of skilled professionals to meet the increasing demand of technology-based businesses and organizations. Therefore, there is an urgent need to simplify and optimize the process of assessing and matching IT talent with the right opportunities.

Traditionally, recruitment and assessment of IT talent is labor-intensive, time consuming, and often subjective. Companies have struggled to efficiently filter through large numbers of candidates to find individuals with the necessary skills and competencies. Additionally, the surge in the number of remote jobs requires innovative solutions for evaluating candidates remotely.

Technological advances have opened the door to effectively address these challenges. The emergence of artificial intelligence or machine learning, especially intelligent systems based on chatbots and Large Language Models (LLM), such as ChatGPT, can pave the way for automating recruitment interviews. This technology enables real-time evaluation of a candidate's skills, talents and compatibility with the job role.

In this context, this paper aims to leverage the technological advances such as LLM to create an innovative solution that will streamline and improve the IT talent recruitment and assessment process. The focus of this project is to develop an LLM-based automated interview system that can assess candidate skills, knowledge and competencies quickly and accurately.

II. LITERATURE REVIEW

A. GPT

GPT (Generative Pre-trained Transformer) is a family of language models developed by OpenAI. GPT is a decoder-only transformer model, which uses masked self-attention [1].

GPT-4 is the latest version of GPT that is publicly available. The GPT model is used as a language model that aims to predict the next token in an incomplete sequence of tokens, based on the previous tokens. The predicted token is then added to the sequence of words and fed into the GPT as input to predict the next token. This is done until the GPT deems the sequence of tokens to be complete. In other words, the GPT model takes several documents and continues their text in the best way in relation to that document. GPT has various interesting applications such as the generation of fictions and poems. GPT does have an advantage in terms of Natural Language Generation (NLG).

B. BERT

BERT (Bidirectional Encoder Representations from Transformers) is designed to pretrain deep bidirectional representations of unlabeled text by conditioning right and left contexts together on all layers. The BERT model architecture is multi-layer bidirectional transformer encoder. BERT can accept input with the maximum token length of 512 [2]. Bert uses masked language modelling technique. Masking is done on 15% of the words in the input document or corpus, and the model is asked to predict the missing words. BERT is one of

the best models for tasks like natural language analysis and sentiment analysis [3].

C. Longformer

Longformer is a BERT-like model which is started from the RoBERTa checkpoint. Longformer can accept input with the maximum token length of 512 (8 times the maximum token length of BERT). Unlike more common Transformer models, sparsifying is done on the full self-attention matrix based on the attention pattern. Unlike full self-attention, the attention pattern on Longformer scales linearly with the input sequence, making it more efficient for longer sequences [4].

D. GLiNER

Traditional NER (Named Entity Recognition) models are effective but are limited to predefined entity types. On the other hand, Large Language Models can extract arbitrary entity types with natural language instruction, offering greater flexibility. However, the cost and size, especially of LLM which can be accessed through the API like ChatGPT, makes LLM unpractical in limited resources scenarios. GLiNER (Generalist Model for Named Entity Recognition using Bidirectional Transformer) is a NER model that can be used in a zero-shot manner. GLiNER uses bidirectional transformer encoder, to facilitate entity extraction parallelly, which is an advantage compared to sequential token generation of LLM. Through comprehensive testing, GLiNER demonstrated good performance, beating ChatGPT and fine-tuned LLM in zero-shot evaluation on various benchmarks [5].

E. Behavioral Event Interview

Behavioral event interview (BEI in abbreviation) is an interview technique which is related to the analysis of the past. This technique puts emphasis on facts and examples from real situations. The core idea of BEI is past behavior can predict future behavior. BEI combines 2 aspects which have been proven significant in increasing the validity and reliability levels of an interview. According to statistical data, the two most important elements which result in higher efficiency of interviews are the structured format and questions which are based on past behavior [6].

The STAR method is often used when conducting BEI. STAR is the abbreviation of Situation, Task Action, and Result. Explanations of the STAR method are the following [7]:

1. *Situation*: The interviewer asks candidates to describe specific situations or contexts they have encountered in the past.
2. *Task*: The interviewer asks about the candidate's role or tasks in the specified situation.
3. *Action*: The interviewer asks for specific actions the candidate took to achieve the goals in the role or task in that situation.
4. *Result*: The interviewer asks about the results of the action that has been carried out.

F. G-Eval: Natural Language Generation Evaluation

G-Eval is a framework which uses LLM with chain-of-thoughts (CoT) to evaluate generated text using the form-filling paradigm. The output from the evaluator is formatted as a form. Experimental results showed G-Eval can outperform other NLG evaluators by a large margin in terms of correlations with human evaluation [8].

G-Eval is a prompt-based evaluator with 3 main components:

1. A prompt containing the definition of the evaluation task and the desired evaluation criteria.
2. A chain-of-thought containing a set of intermediate instructions generated by the LLM that explains the detailed evaluation steps.
3. A scoring function that calls the LLM and calculates a score based on the probabilities of the token returned.

To get a more detailed and continuous score which mirrors the quality and diversity of the generated text, G-Eval uses the probabilities of output tokens to normalize the scores and take a weighted sum as the result. Formally, if exists a set of scores $S = \{s_1, s_2, \dots, s_n\}$ which has been defined at the prompt, probability of each score $p(s_i)$ is calculated by the LLM, the final score is calculated as in (1).

$$score = \sum_{i=1}^n p(s_i) \times s_i \quad (1)$$

G. Natural Language Generation Evaluation Metrics

Natural Language Generation (NLG) is a multifaceted problem which requires assessment of various desired criteria, such as fluency, coherency, coverage, relevance, adequacy, overall quality, etc. A set of popular criteria is proposed for various popular NLG tasks [9].

The various tasks are machine translation, question generation, abstractive summarization, dialogue generation, image captioning, data to text generation, all the previous tasks. Tasks that are related to an interview bot are question generation and dialogue generation. The criteria used in this paper are:

1. *Relevance*: Metric to determine whether the generated question is related to the source material they are based upon. This is a criterion for question generation.
2. *Answerability*: Metric to determine whether the generated question is answerable given the context. This is a criterion for question generation.
3. *Listening*: Metric to determine whether the bot pays attention to what the user says. This is a criterion for dialogue generation.

III. RELATED WORKS

In a prior work, the development of an interview bot is also carried out using BEI technique with the STAR method. This approach is carried out for interviews in the Indonesian language. This approach is carried out with NLP and machine learning [10].

IV. PROPOSED SOLUTION

A. Interview Generation

The system will conduct interviews for each job competency provided by the recruiter. Each job competence will be assigned individually to the LLM. LLM will be given an order to conduct an interview related to that competency using the BEI technique using the STAR method first (behavioral interview), then LLM will be given an order to generate one technical question for that competency (technical interview), if the LLM deems that the competence is a technical competence. The flow diagram of the interview can be seen on Fig. 1.

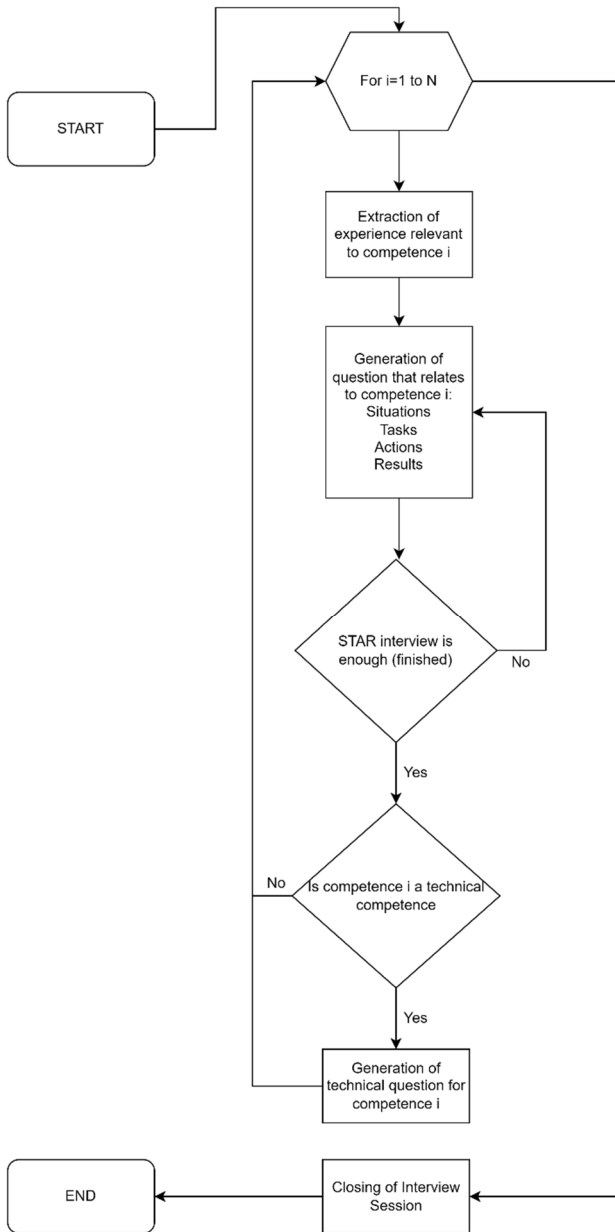


Fig. 1. Flow diagram of the interview.

B. Behavioral Interview Question Generation

To generate questions asked in the STAR interview, few-shot learning is carried out in LLM through prompting. The data used in few-shot learning is 2 data. When generating questions, LLM will receive the competencies that must be tested so in the data used in few-shot learning there are examples of competencies that are being tested for examples in the data.

In the STAR interview, to find out if the job competence has been answered sufficiently, once every interaction (one question answered) is completed, the job competence that is being asked (competence i) and the history of all interactions for this competence will be given to an LLM. This LLM will determine whether the interviews for this competency (competence i) have been carried out sufficiently or not. If not, the LLM will be given instructions to generate further questions in the competence interview. If the interview for a competence has been completed. The system will repeat the interview for the next competence (competence $i + 1$) until all

the competences is exhausted. After that, the system will close the interview session. In generating questions, the LLM will be given relevant experience with the competencies being tested which comes from the as reference material.

C. Technical Interview Question Generation

Generating technical interview questions is done by doing few-shot learning on LLM. The data used in few-shot learning comes from the book Python Interview Questions by Swati Saxena. The book only contains questions that are relevant to the Python programming language. At the prompt, the LLM will be instructed to generate questions appropriate to the technical skill being tested (which may not be Python) based on the examples given in the few-shot learning.

If the competency provided is not a technical competency (technical questions cannot be generated for that competency), LLM will be given an order to provide a certain output so that the system continues the interview to the next stage without providing technical interview questions for that competency.

The amount of data used in few-shot learning is 5 data. Of the five data, 2 are technical questions that display a section of code and ask for the output of that code and 3 are technical questions that ask about other aspects, for example regarding libraries, modules or built-in functions of a tech stack. LLM will be notified in the prompt that the examples in the few-shot prompt only apply to the Python programming language.

D. Interview Evaluation or Scoring

Interviews that have been conducted must be evaluated to determine whether the interviewee's answers are in accordance with the competency being tested or not. The scoring will be done for each competence separately, this can be done since the question generation is also separate for each competence. The flow diagram of the interview evaluation or scoring can be seen in Fig. 2.

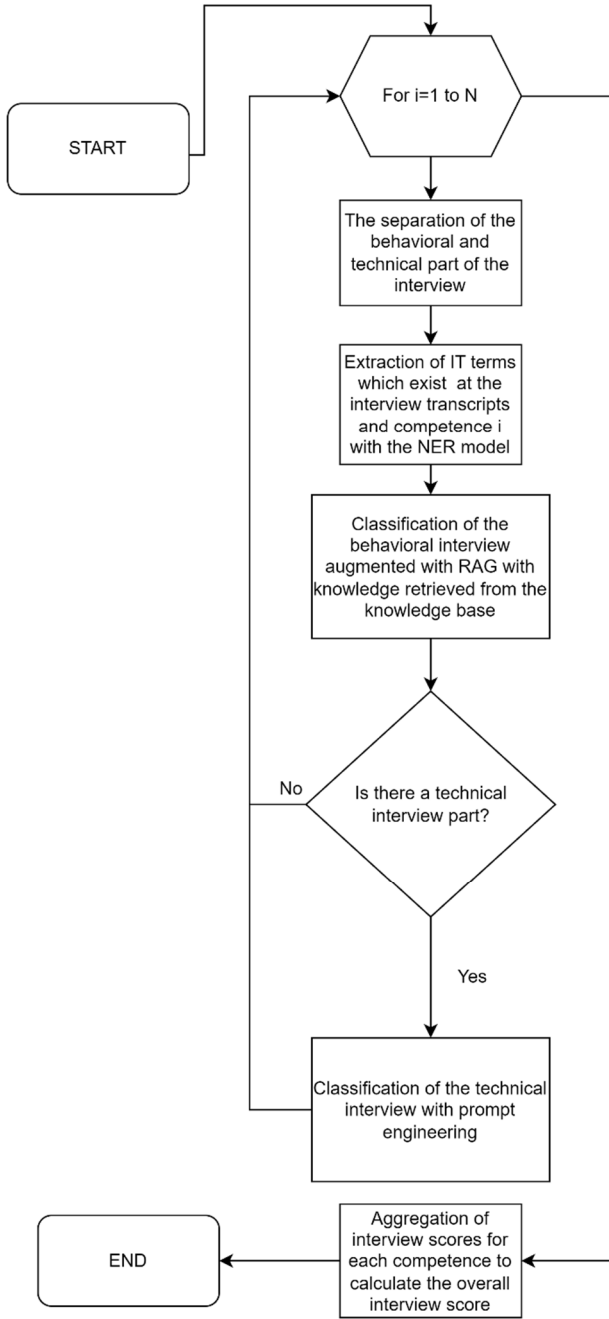


Fig. 2. Flow diagram of interview evaluation or scoring.

E. Behavioral Interview Evaluation or Scoring

The evaluation will be treated as a binary classification problem. The 2 labels or classes for this classification are successful interview and failed interview. A fine-tuned model will be used for the classifier.

To carry out fine-tuning of the model required in this stage, data from interview transcripts conducted using the STAR method, the competency being tested, and the evaluation results of the transcript regarding that competency are needed. Due to the unavailability of data relevant to this evaluation stage, synthetic data will be generated using LLM. The LLMs that will be used in this synthetic data generation stage are GPT 4 and GPT-4o.

Input to the evaluation model will also be augmented with knowledge originating from the IT domain knowledge base.

The knowledge base used is the first 7 sentences of the summary section in the Wikipedia article related to the IT term.

IT terms will be extracted from conversation transcripts and competencies tested with the NER model. The NER model used in this case is GLiNER. This model is used because it has sophisticated capabilities and can be used in a zero-shot manner making it convenient to use. The NER model is simply given the types of entity that wants to be extracted, and the model will identify these entities in the text. There are 3 types of entities used in the NER model for this case, namely software package, programming language, and software tool.

The input to the Longformer model consists of 3 parts:

1. **KNOWLEDGE:** Filled with knowledge of IT terms. The IT terms are extracted from the COMPETENCE and TRANSCRIPT part.
2. **COMPETENCE:** The competence being evaluated (the i -th competence in Fig 2).
3. **TRANSCRIPT:** The interactions between the interviewer and interviewee for the i -th competence.

F. Technical Interview Evaluation or Scoring

Technical evaluation will be carried out by giving orders to the LLM to determine whether the interviewee's answers are correct or not. To help LLM determine the evaluation results, several-shot learning will be carried out. The data used in few-shot learning are the same as the few-shot data used in the technical interview question generation. The book only contains questions that are relevant to the Python programming language. At the prompt, the LLM will be instructed to carry out an evaluation appropriate to the technical skill being tested (which may not be Python) based on the examples given in the few-shot learning.

G. Interview Score Calculation

Scores will be calculated for each competence first, (each competence i as in Fig. 2). There are 2 formulas used for calculating the score of interviews for each competence. If the competence is a technical competence, (2) is used. If the competence is not a technical competence, (3) is used. Competence score for the i -th competence is denoted with c_score_i , behavioral score for the i -th competence is denoted with b_score_i , technical score for the i -th competence is denoted with t_score_i . Behavioral and technical score only have 2 possible values since they are the result of binary classifications (1 for successful interview and 0 for failed interview).

$$c_score_i = \frac{b_score_i + t_score_i}{2} \quad (2)$$

$$c_score_i = b_score_i \quad (3)$$

After the scores for each interview are calculated, the scores are collected and used to count the average score of each competence score. This average is the final score of the interview. The formula for calculating the interview score is (4) where N denotes the number of competences. This interview score has values ranging from 0 to 1. If a maximum interview score of 100 is desired, for example, simply multiplying the interview score with the desired maximum score (for example 100) will scale the interview score accordingly.

$$\text{interview_score} = \frac{\sum_{i=1}^n c_score_i}{N} \quad (4)$$

V. EXPERIMENT AND TESTING

A. Longformer Fine-tuning

The dataset is created using the initial dataset as a source of information. The initial dataset is Real / Fake Job Posting Prediction released by Shivam Bansal. The dataset was created with the aim of classifying real and fake job descriptions. In this project, only original data is used. Here are the steps in creating the dataset:

1. From this dataset, jobs related to the IT sector are extracted by taking data only for those that have the value of "Information Technology" in the "function" column.
2. The "requirements" column is then extracted from the data and entered into the LLM prompt. At this prompt, few-shot learning is carried out and the LLM is given the command to extract existing competencies in list form to become new data.
3. The "title" column is also taken from the data and entered the LLM prompt. At this prompt, several-shot learning is carried out and the LLM is given the command to create a job description for that job title that corresponds to each competency that has been extracted in the previous stage.
4. The data in the previous stage is entered into the LLM prompt. In this prompt, several-shot learning was carried out and LLMs were given instructions to create interview interactions in the STAR format with half the interviews failing and half the interviews being successful. In creating failed interview interactions, LLM is given orders to create failures from interviewees at different stages (stage S, T, A, or R).

This data will be used in fine-tuning the STAR interview evaluation (Longformer) model. The total amount of data is 300 with 240 data (80%) as training data and 60 data (20%) as test data. The LLM model used in creating this dataset is the GPT-4 and GPT-4o model from OpenAI.

At the start of the model fine-tuning experiment, the base BERT model was used initially. Data tokenization was carried out with the BERT tokenizer. The token resulting from this tokenization has a maximum length of 1068, an average length of 572.33, and a median length of 546.0. The number of tokens that have a length of more than 512 is 159 data. This number is quite a large percentage of the total amount of data of 300. The Longformer model is used as the behavioral interview evaluation model because Longformer can handle a greater number of input tokens than BERT (BERT has a maximum number of tokens for input of 512).

The metric used in fine-tuning is F1-score. After fine-tuning was carried out, the following metric scores were obtained:

- *F1-score*: 0.9615.
- *Accuracy*: 0.9667.
- *Precision*: 0.9691.
- *Recall*: 0.9667.

B. NLG Evaluation with G-Eval

The evaluation process with G-Eval is done using the DeepEval library in the Python programming language. The library only allows chain-of-thoughts prompting to carry out the evaluation. The LLM used is GPT-4.

A metric to check whether the series of questions generated by the behavioral question generator component are in accordance with the STAR structure or not is also made, this is called STAR Completeness. This can be done due to the flexibility of the G-Eval framework.

The chain-of-thought prompt for the answerability metric is in Table I. The chain-of-thought prompt for the relevance metric is in Table II. The chain-of-thought prompt for the listening metric is in Table III. The chain-of-thought prompt for the STAR completeness metric is in Table IV.

TABLE I. CHAIN-OF-THOUGHT PROMPT FOR THE ANSWERABILITY METRIC

Prompt
Read the measured competence, relevant experience (could be none), and conversation carefully
The conversation could still be empty (it just started)
Determine whether the last generated question (the output) by the interviewer is answerable given the competence and previous conversation

TABLE II. CHAIN-OF-THOUGHT PROMPT FOR THE RELEVANCE METRIC

Prompt
Read the measured competence, relevant experience (could be none), and conversation carefully
The conversation could still be empty (it just started)
Determine whether the last generated question by the interviewer (the output) is related to the measured competence it is based upon

TABLE III. CHAIN-OF-THOUGHT PROMPT FOR THE LISTENING METRIC

Prompt
Read the measured competence, relevant experience (could be none), and conversation carefully
The conversation could still be empty (it just started)
You need to determine from the last generated question whether the interviewer pay attention to the conversation
Determine whether the last generated question by the interviewer is related to the previous conversation

TABLE IV. CHAIN-OF-THOUGHT PROMPT FOR THE STAR COMPLETENESS METRIC

Prompt
Read the measured competence, relevant experience (could be none), and conversation carefully
You need to determine whether the questions generated by the interviewer follow the STAR (Situation, Task, Action, and Result) structure
Each question should correspond to one letter on the "STAR"
It is okay if the questions do not fully end at Result (STAR), it is okay if it ends in Situation (S), it is okay if it ends in Task (ST), and it is also okay if it ends in Action (STA)

Evaluation with G-Eval is done using the first 50 data from the 300 synthetic data generated for fine-tuning. The evaluation result with G-Eval can be seen in Table V.

Here are the scenarios for G-Eval evaluation (for each one of the 50 data):

1. The behavioral interview question generator is given one competence.
2. The question generator generates a question.
3. The question generator is given the answer to that question from the data.
4. Steps 2 and 3 are repeated until there is no answer any more on the data or the interview generator deems the interview finished.
5. All the questions generated for this competence are collected to measure the STAR completeness metric.

TABLE V. G-EVAL EVALUATION RESULT

Metric Name	Metric Value
Answerability	0.8938
Relevance	0.9860
Listening	0.9633
STAR Completeness	0.9502

An example of interaction between interviewer (interview bot) and the interviewee (user/candidate) during STAR interview can be seen in Fig. 3.

COMPETENCE: Experience with server-side MVC frameworks, preferably Ruby on Rails

INTERVIEWER:
Can you describe a situation where you had to leverage your experience with Ruby on Rails for a critical project?

INTERVIEWEE:
Certainly. At my previous job, I was tasked with leading the development of a server-side application using the Ruby on Rails MVC framework. The project involved creating a complex, high-performance system to manage a large database of customer information and ensure seamless interaction with the frontend.

INTERVIEWER:
What specific tasks were you assigned in this project, and what were the main objectives you aimed to achieve?

INTERVIEWEE:
I was responsible for designing the overall architecture of the backend system, ensuring scalability and reliability. My main objective was to create a robust and efficient backend that could handle a high volume of requests without compromising on performance.

INTERVIEWER:
What actions did you take to implement and maintain the Ruby on Rails application, and how did you ensure its robustness and efficiency?

INTERVIEWEE:
I started by clearly defining the models, views, and controllers to ensure a clean separation of concerns. I used `ActiveRecord` for database interactions, implemented caching mechanisms, and optimized SQL queries to boost performance. Regular code reviews and testing were conducted to maintain code quality. Additionally, I utilized background jobs to handle time-consuming tasks asynchronously.

INTERVIEWER:
What was the outcome of your implementation, and how did it impact the overall project?

INTERVIEWEE:
The project was a success. The backend system performed exceptionally well, handling thousands of requests per second without any significant downtime. Our client was very satisfied with the result, and this enhanced our reputation as a reliable development team. The project showcased our ability to deliver robust, scalable, and efficient server-side solutions using Ruby on Rails.

Fig. 3. An example of interaction between interviewer and interviewee during STAR interview.

VI. CONCLUSION

By giving the GPT-4o LLM instructions at the initial prompt to generate STAR-based questions and providing examples of conversations with a small number of STAR structures with few-shot learning, LLM has succeeded in generating questions with STAR structure. The test results with G-Eval showed very good results with a listening metric value of 0.9633 and a STAR completeness metric value of 0.9502.

By giving the GPT-4o LLM examples of conversations with a relatively small number of tested competencies in prompts with few-shot learning, LLM has succeeded in generating questions with a STAR structure that are relevant

to the questions being tested. The use of competency examples tested on example data in few-shot learning produced very good G-Eval test results with a relevance metric value of 0.9924 and an answerability metric value of 0.8938.

The use of the Longformer model, which is a BERT-based model, to classify synthetic IT talent interview conversation data was successful. A BERT-based model that uses a deep bidirectional representation of text by conditioning left and right contexts together has proven to be suitable for NLU (Natural Language Understanding) problems. The use of RAG using GLiNER in classification with a Wikipedia knowledge base as a source of IT-related knowledge can be used with BERT-based models. The Longformer model is used because adding knowledge obtained from RAG requires a longer number of tokens than the regular BERT model. The evaluation results on synthetic data show very good results with an F1-score of 0.962, an accuracy value of 0.9667, a precision value of 0.9691, and a recall value of 0.9667.

REFERENCES

- [1] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding with Unsupervised Learning," Technical Report, OpenAI, 2018.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," arXiv (Cornell University), Jan. 2018, doi: 10.48550/arxiv.1810.04805.H. Simpson, *Dumb Robots*, 3rd ed., Springfield: UOS Press, 2004, pp.6-9.
- [3] B. Ghogjogh, A. L. I. Ghodsi, and U. Ca, "Attention Mechanism, Transformers, BERT, and GPT: Tutorial and Survey," ArXiv, vol. 1, no. 1, pp. 1–14, 2017, doi: 10.31219/osf.io/m6gc.
- [4] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The Long-Document Transformer," arXiv (Cornell University), Jan. 2020, doi: 10.48550/arxiv.2004.05150.
- [5] U. Zaratiana, N. Tomeh, P. Holat, and T. Chamois, "GLiNER: Generalist Model for Named Entity Recognition using Bidirectional Transformer," arXiv (Cornell University), Jan. 2023, doi: 10.48550/arxiv.2311.08526.
- [6] T. Raisová, "The Comparison Between the Effectiveness of the Competency Based Interview and the Behavioral Event Interview," 2012.
- [7] J. E. Sharp, "Behavioral Interview Training in Engineering Classes," presented at the 2012 ASEE Annual Conference & Exposition, San Antonio, TX, USA, Jun. 2012, doi: 10.18260/1-2--21011.
- [8] Y. Liu, D. Iter, Y. Xu, S. Wang, R. Xu, and C. Zhu, "G-Eval: NLG Evaluation using GPT-4 with Better Human Alignment," arXiv.org, Mar. 29, 2023. <https://arxiv.org/abs/2303.16634v3>.
- [9] A. B. Sai, T. Dixit, D. Y. Sheth, S. Mohan, and M. M. Khapra, "Perturbation CheckLists for Evaluating NLG Evaluation Metrics," *arXiv*, 2021. doi: 10.48550/arxiv.2109.05771.
- [10] J. Siswanto, S. Suakanto, M. Andriani, M. Hardiyanti, and T. F. Kusumasari, "Interview Bot Development with Natural Language Processing and Machine Learning," *International Journal of Technology*, vol. 13, no. 2, p. 274, Apr. 2022, doi: 10.14716/ijtech.v13i2.5018.