

# AI-Enhanced Interview and Assessment System

## **Team Members:**

Kritika Acharya(THA077BCT024)

Rashna K.C.(THA077BCT039)

Suyog Acharya(THA077BCT045)

Yogesh Aryal( THA077BCT046)

## **Supervised By:**

Er. Sudip Rana

Department of Electronics and Computer Engineering

Institute of Engineering, Thapathali Campus

August, 2024

# Presentation Outline

- Motivation
- Objectives
- Project Scope
- Project Applications
- Methodology
- Results
- Analysis of Results
- List of Remaining Tasks
- References

# Motivation

- Traditional interviewing is resource-intensive, time-consuming, and dependent.
- Create fairer, consistent recruitment processes through AI.
- AI interviews minimize bias and focuses on skills.
- Conduct Interviews continuously without human intervention.

# Objectives

- To automate the technical interview process, reducing manual effort and ensuring unbiased evaluations.
- Implement T5, WaveNet, Whisper, OpenCV for precise assessments.

# Project Scope

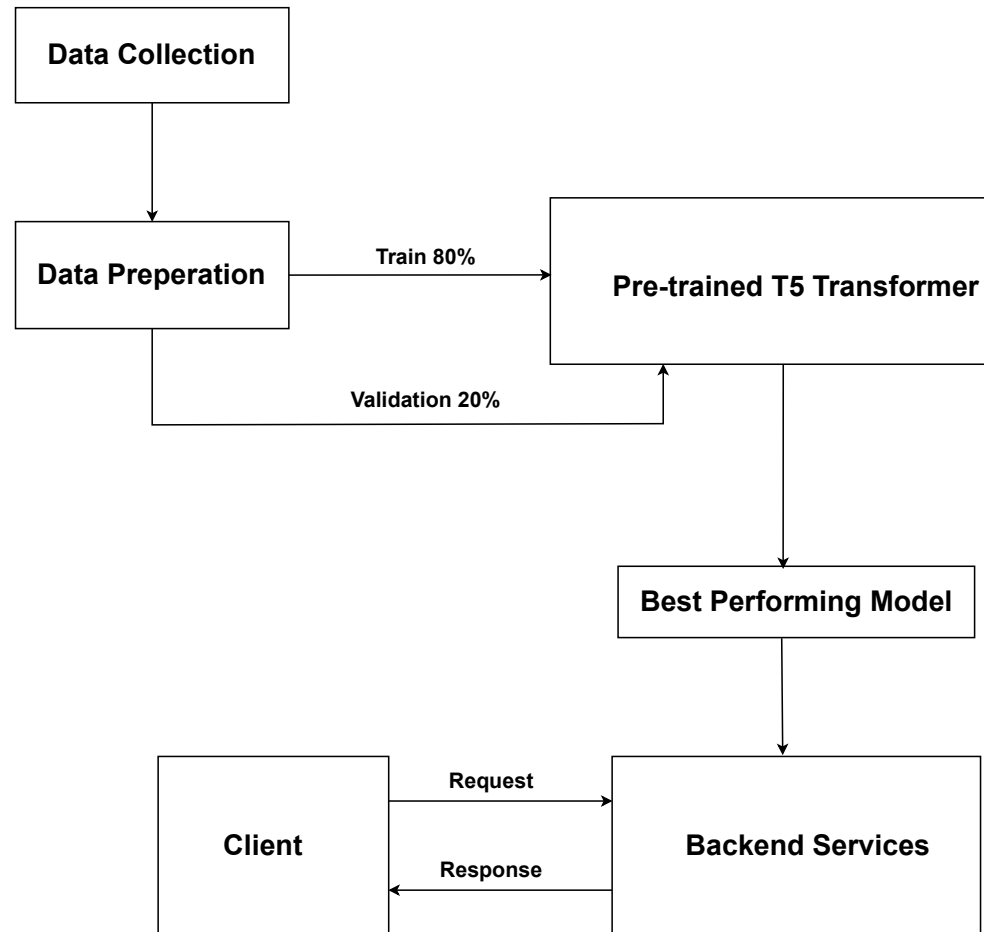
- Transform the hiring process with consistency and efficiency.
- Ensure a uniform interview process, minimizing bias and error.
- Handle large volume of applicants without degradation of performance.

# Project Applications

- **Recruitment and Hiring:**
  - Unbiased Evaluations
  - Scalability
- **Business Efficiency:**
  - Cost Saving (32.7%)
  - Resource Optimization
- **Educational Sector:**
  - Practice Interviews
- **Research:**
  - Behavioral studies
  - Data collection

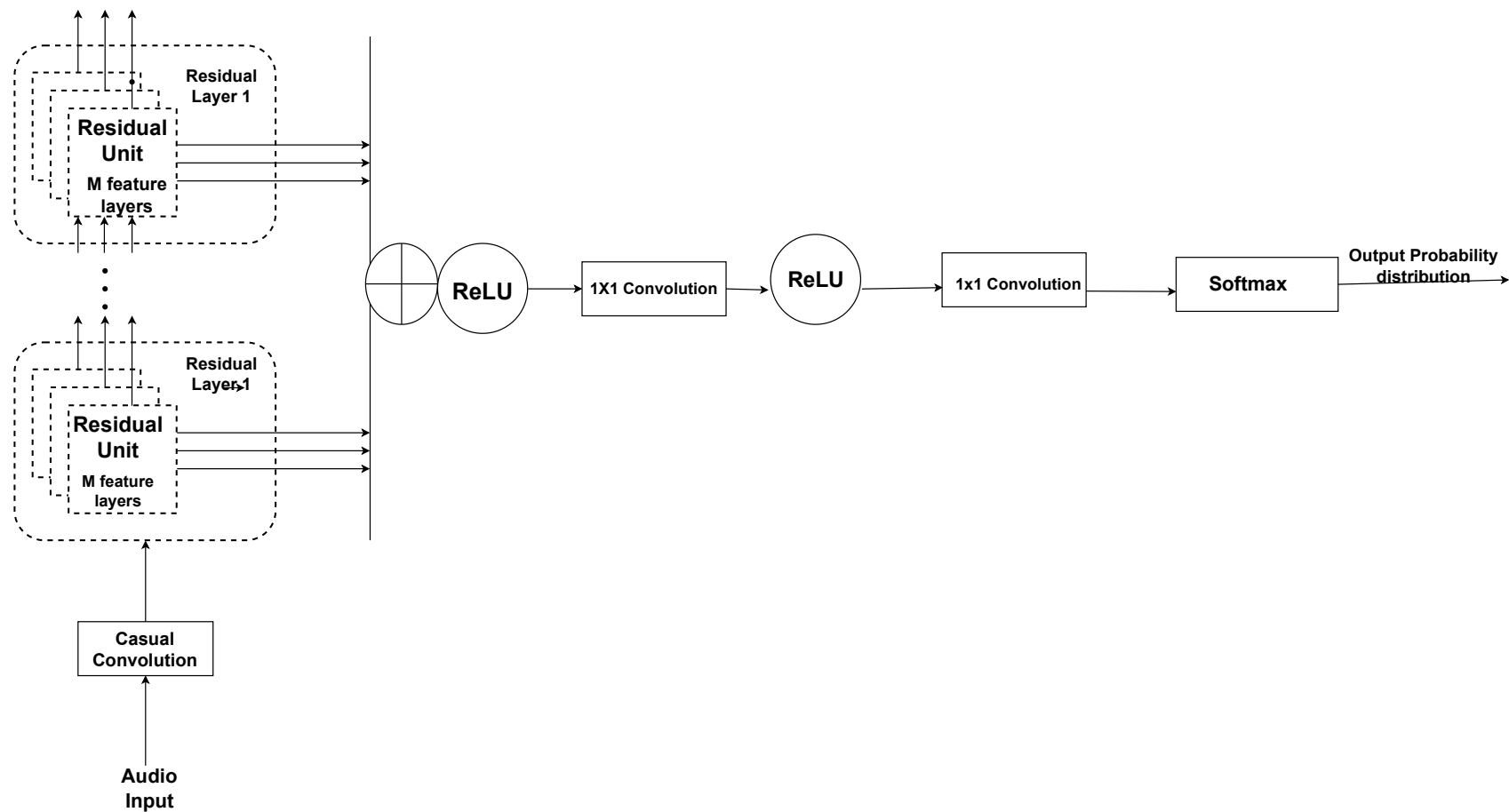
# Methodology - [1]

## (System Block Diagram)



# Methodology – [2]

## WaveNet Text-to-Speech





# Methodology – [3]

## WaveNet Working principle I

- **Autoregressive and Probabilistic Model:** Conditions each audio sample on all previous samples for accuracy.
- **High-Quality Speech Generation:** Produces natural and contextually relevant speech output from text inputs.
- **Linguistic Feature Conditioning:** Enhances speech relevance by conditioning on linguistic features from input text.
- **Speaker Identity Conditioning:** Accurately captures and reproduces characteristics of different speakers.
- **Extensive Training Data:** Trained on thousands of audio samples per second for nuanced learning.

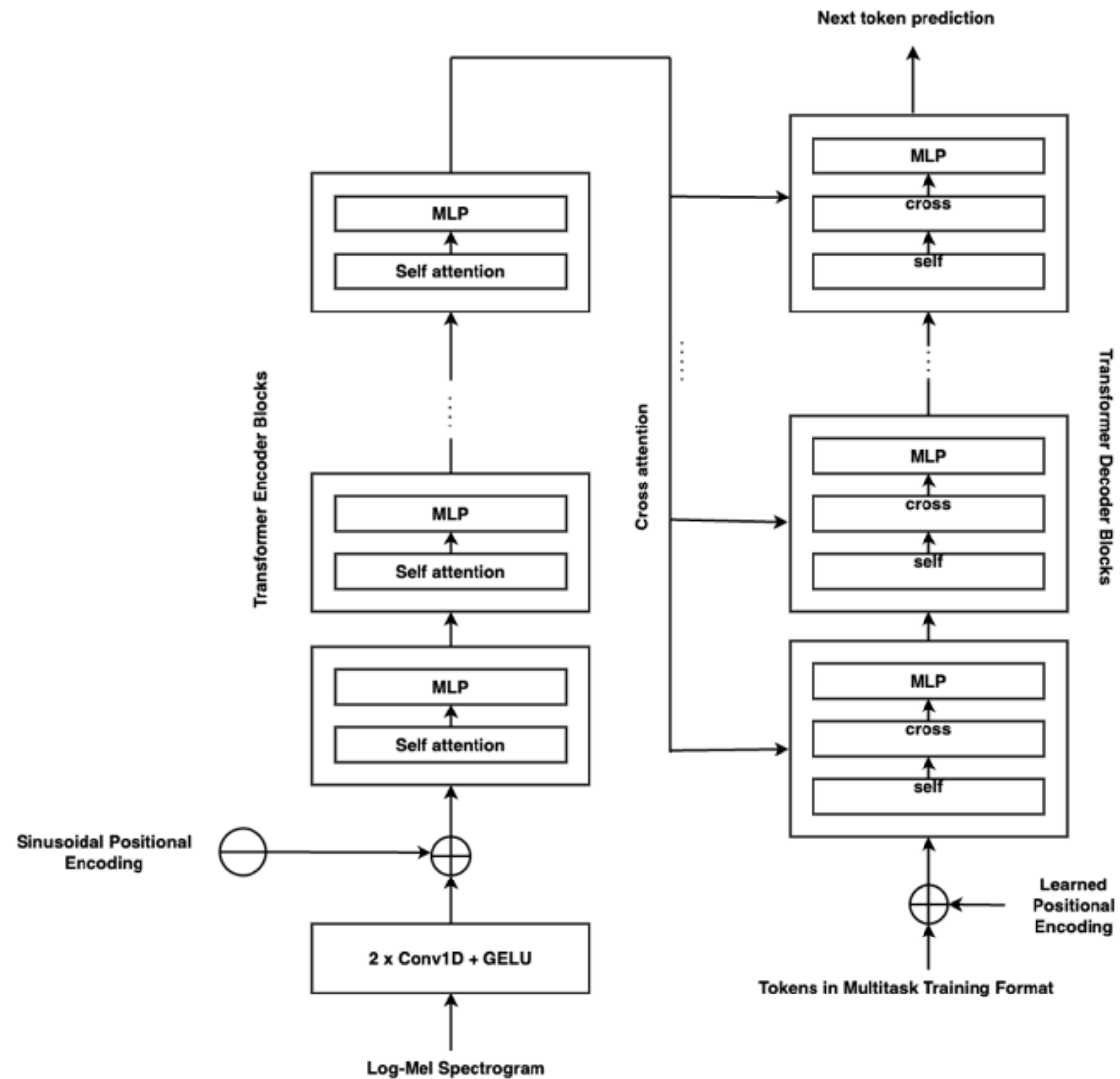
# Methodology – [4]

## WaveNet Working principle II

- **Dilated Convolutions:** Expands receptive fields exponentially, capturing long-range temporal dependencies in audio.
- **Efficient Training:** Handles large audio datasets effectively, optimizing performance.
- **Deep Generative Architecture:** Directly models raw audio waveforms for high-fidelity speech synthesis.

# Methodology – [5]

## Whisper Speech-to-Text



# Methodology – [6]

## Whisper Working principle I

- **Innovative Encoder-Decoder:** Efficiently captures accurate text from audio.
- **Self-Attention Mechanism:** Allows sequence parts to communicate, capturing long-range dependencies.
- **MLP Layers:** Uses feed-forward neural networks to transform input audio sequences effectively.
- **Normalization Layers:** Ensures model stability through regulation and normalization processes.
- **Complex Relationship Learning:** Captures long-range dependencies between input elements.

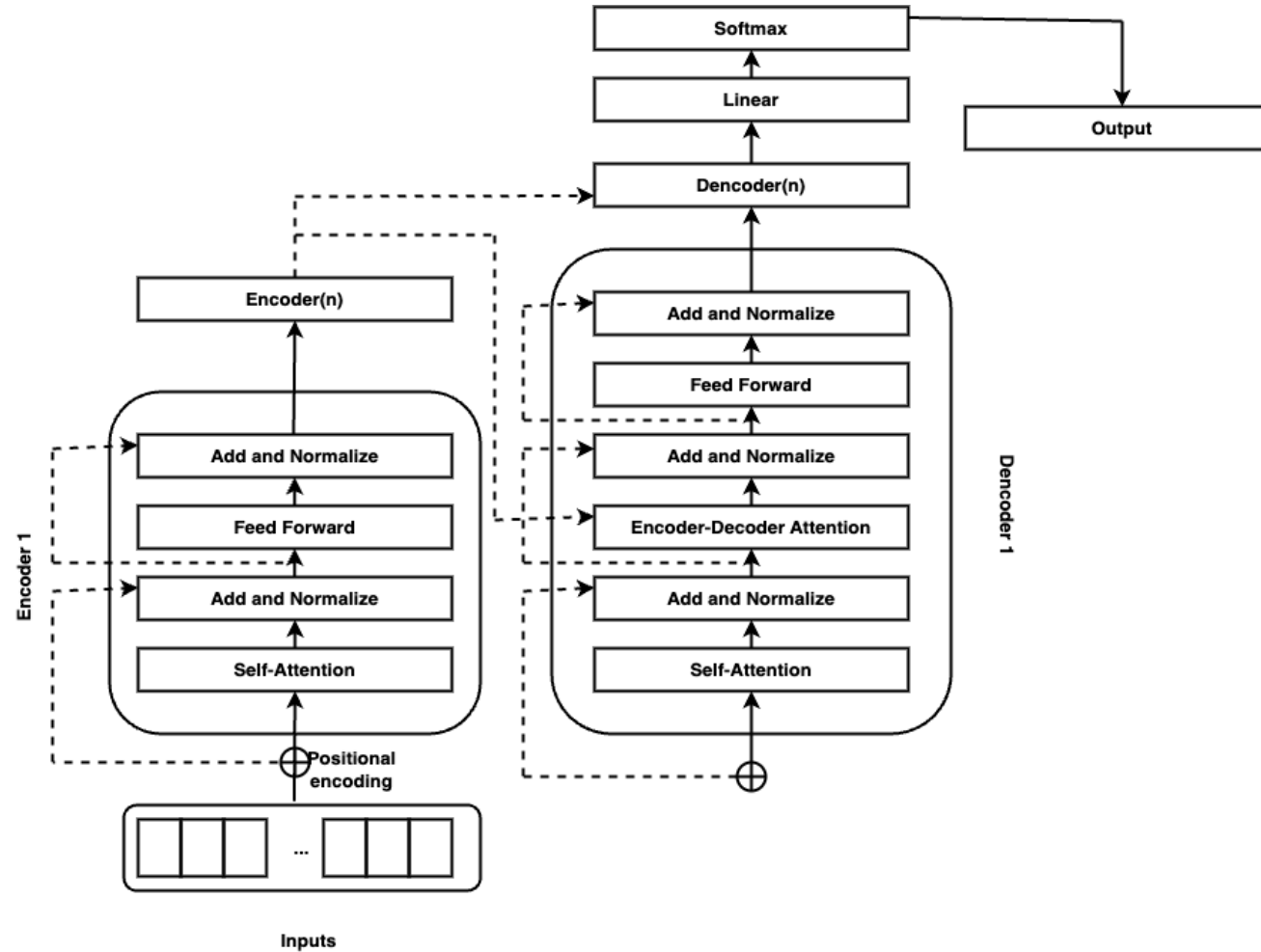
# Methodology – [7]

## Whisper Working principle II

- **Cross-Attention Mechanism:** Focuses on specific input parts using contextual information from the encoder.
- **Residual Connections:** Utilizes strengths of all components to create accurate, coherent subtitles.
- **Special Tokens:** Directs tasks like language identification, timestamps, multilingual transcription, and translation.

# Methodology – [8]

## T5 for Conditional Generation



# Methodology – [9]

## T5 Working principle I

- **Inputs**

- Textual inputs are tokenized and embedded
- Positional encoding is added to the input embeddings

- **Encoder Stack**

- Consists of multiple identical layers
- Each layer has:
  - Self-Attention Mechanism to allow each position in input sequence to attend to all other positions
  - Add and Normalize Layer to combine input and output of the preceding layer
  - Feed-Forward Network to apply a point-wise feed-forward network to each position separately

# Methodology – [10]

## T5 Working principle II

- **Decoder Stack**

- Consists of multiple identical layers
- Each layer has:
  - Self-Attention Mechanism to attend to the decoder's previous outputs
  - Add and Normalize Layer
  - Encoder-Decoder Attention Mechanism to allow the decoder to attend to the encoder's outputs

- **Output Layer**

- Applies a linear layer followed by a Softmax layer to generate probabilities over the vocabulary



# Methodology – [11]

## T5 Implementation I

- Initialized T5 tokenizer and model (t5-small) from Hugging Face.
- Input and target texts were tokenized, handling padding and truncation.
- Defined training arguments: evaluation strategy, batch size, epochs, saving steps.
- Initialized Trainer with model, training arguments, datasets, data collator, tokenizer.
- Model was trained using the Trainer.
- Trained model and tokenizer were saved.

# Methodology - [12]

## T5 Implementation II

- **Model Loading:** Loads T5 model and tokenizer from specified path for question generation.
- **Configuration:** Defines interview categories, time limits, difficulty levels, and scores.
- **Question Management:** Tracks asked questions, defines subcategories for each main category.
- **Generate Unique Questions:** Generates questions with T5 by category, difficulty, and answers.
- **Evaluate Answers:** Placeholder function to score user answers.
- **Format Time:** Converts time into minutes and seconds for display.

# Methodology - [13]

## T5 Implementation III

- Model dynamically adjusts question difficulty and relevance based on responses.
- We'll keep collecting data and fine-tuning the model for optimal hyperparameters.
- After implementing scoring mechanism based on keywords matching, this model will be ready for further implementations.

# Methodology - [14]

## Dataset Overview

- Total number of dataset:
  - Technical questions dataset: 2731 (Includes Question from topics like Html, Css, JavaScript, Node.js, MongoDB, MySQL, API)
  - Behavioral questions dataset: 1086

# Methodology - [15]

## Dataset Overview I

- Example of Technical Interview Question Dataset:

```
{
  "question": "What is npm?",
  "answer": "npm is the default package manager for Node.js, which allows you to install, share, and manage dependencies of Node.js projects.",
  "followup_question": "How do you install a package globally using npm?",
  "difficulty": "easy",
  "topic": "Node.js",
  "keywords": [
    "package manager",
    "dependencies",
    "install"
  ],
  "category": "backend"
},
{
  "question": "What is an event loop in Node.js?",
  "answer": "The event loop is a single-threaded loop that handles all asynchronous callbacks in Node.js. It allows Node.js to perform non-blocking I/O operations.",
  "followup_question": "How does the event loop handle I/O operations in Node.js?",
  "difficulty": "medium",
  "topic": "Node.js",
  "keywords": [
    "event loop",
    "single-threaded",
    "asynchronous callbacks",
    "non-blocking I/O"
  ],
  "category": "backend"
},
```

# Methodology - [16]

## Dataset Overview II

The properties in the JSON objects are:

- **question**: The main query or prompt being addressed.
- **answer**: The detailed response or explanation provided.
- **followup\_question**: A related question exploring deeper into topic.
- **difficulty**: The complexity level: easy, medium, or hard.
- **topic**: The specific subject matter of the question.
- **keywords**: Important terms associated with the question.
- **category**: Broader classification of the question's nature.

# Methodology - [17]

## Dataset Overview III

- Example of Behavioral question dataset:

```
{
  "question": "Are you a self-motivator?",
  "answer": "Absolutely. For me, internal motivation works far more than external motivation ever could.",
  "followUp": "Awesome. How would you spread motivation to others?"
},
{
  "question": "What matters to you more - job satisfaction or salary?",
  "answer": "According to me, job satisfaction covers all - the quality and quantity of work, salary, company environment and others. Yes, cooperation and adjustment is needed in a company. ",
  "followUp": "Are you ready to work in a company who offers you 5 times more than what you get now but the area may not be very interesting to you?"
},
{
  "question": "Have you worked with someone unprofessional, how did you handle it?",
  "answer": "During my B-Tech final semester internship, I have experienced unprofessional behavior. I did not do anything to show my displeasure, instead, I kept behaving professionally. ",
  "followUp": "Do you think not showing the displeasure is the only way to tackle that situation?"
},
```

# Methodology - [18]

## Data Collection Process

- **Web Scraping**

- To collect the technical interview questions, we implemented a web scraping solution.
- We used Selenium WebDriver with Chrome, configuring options for optimal performance and compatibility.
- The script navigates to predetermined URLs containing relevant interview questions.
- Using XPath and CSS selectors, the script locates and extracts question and answer elements from the web page.
- Extracted data is stored as JSON file for further processing.



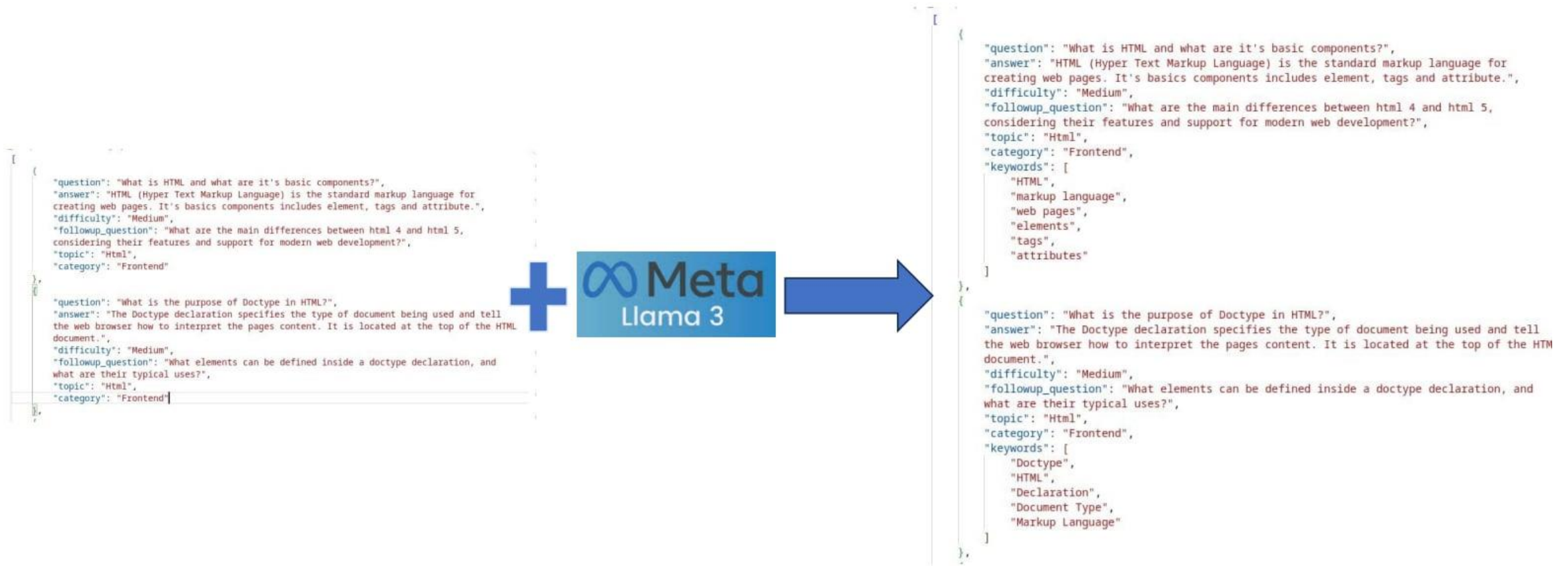
# Methodology - [19]

## Keywords Extraction

- We have implemented a keyword extraction process using the LLAMA 3 model.
- For each question-answer pair, a contextual prompt is created to generate relevant keywords.
- Script uses subprocess to run LLAMA 3 model with generated prompts.
- The model's output is processed to extract 3-5 relevant keywords for each question-answer pair.
- The extracted keywords are added to the existing JSON structure.

# Methodology - [20]

## Keywords Extraction-I



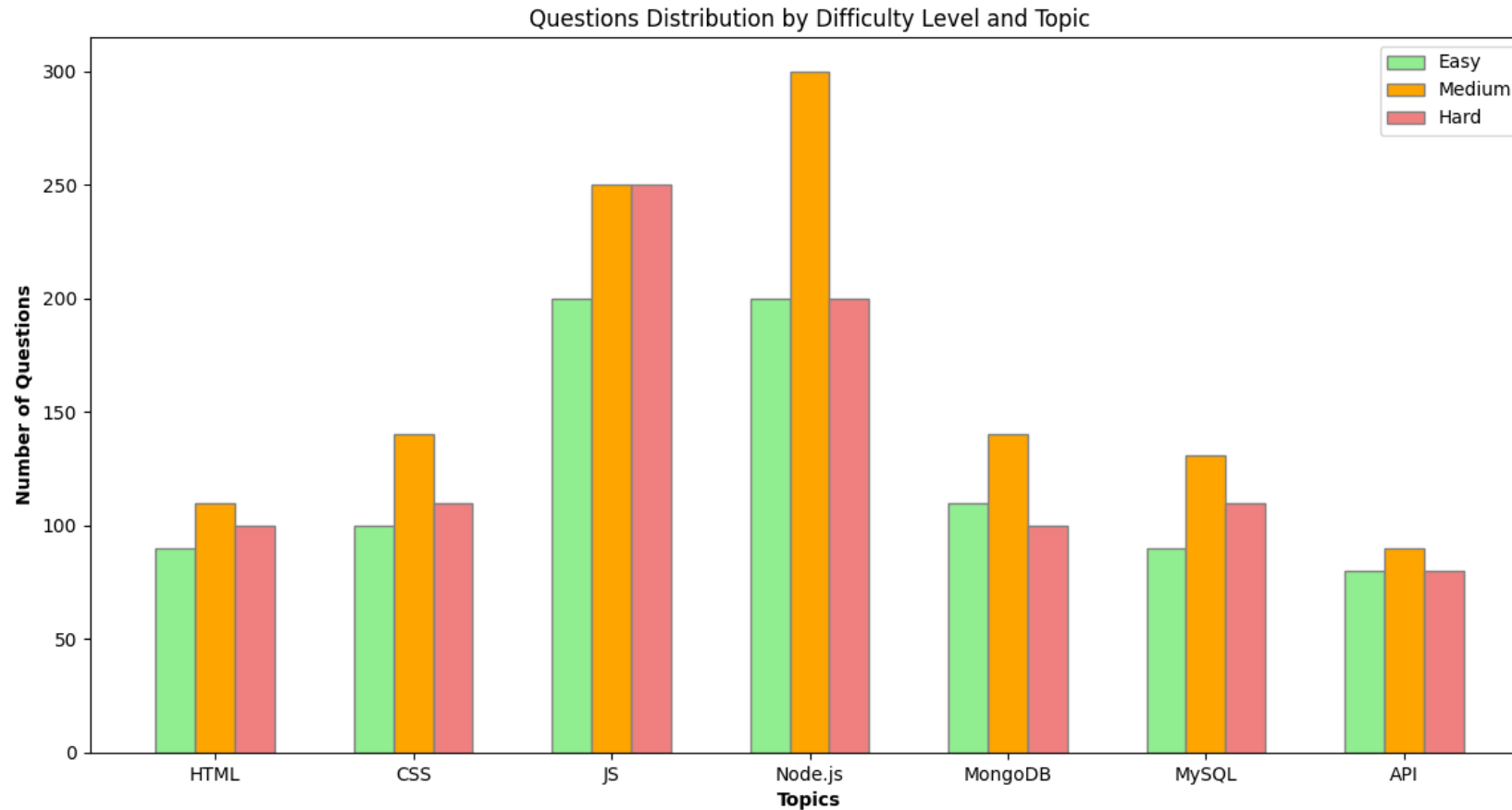
# Methodology - [21]

## Dataset Collection Sources

- **MDN Web Docs** : Supported by major tech companies like Google, Microsoft, Mozilla, Samsung,
- **Official documentation**: Used for preparing MongoDB, MySQL, Node.js dataset.
- **Turing**: Collaborates with companies like Google, Meta to provide remote job opportunities.
- **Flexiple**: It has immense practical experience and industry involvement.
- **GeeksForGeeks**: It is widely regarded in the tech community.

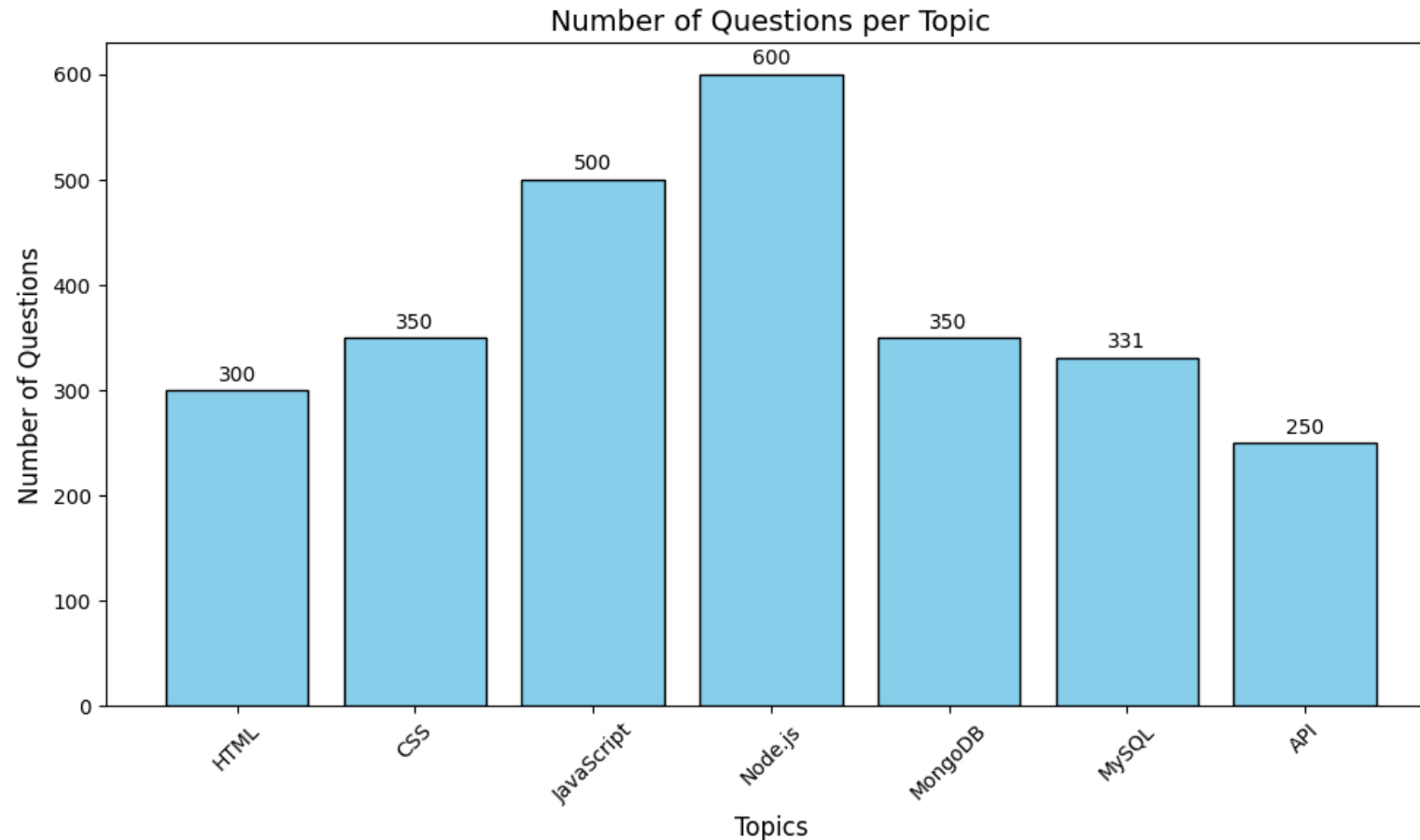
# Methodology - [22]

## Data Visualization



# Methodology - [23]

## Data Visualization II



# Methodology - [24]

## Data Visualization III

### Number of Questions per Topic:

- Node.js: 600 questions (22.56%)
- JavaScript: 500 questions (18.80%)
- MongoDB: 350 questions (13.16%)
- CSS: 350 questions (13.16%)
- MySQL: 331 questions (12.44%)
- HTML: 300 questions (11.28%)
- API: 250 questions (9.40%)

### Distribution across difficulty levels:

- Easy: 30%
- Medium: 40%
- Hard: 30%

# Instrumentation – [1]

## (Hardware Requirements)

### Cloud Computing Resources:

Google Colab:

Hardware Components	Details
CPU	Intel Xeon 2.20 GHz
GPU	NVIDIA T4 GPU
Number of GPUs	1
CUDA cores per GPU	2560
RAM	16 to 30 GB
Memory Bandwidth	320 GB/s

# **Instrumentation – [2]**

## **(Software Requirements)**

- Text Editor: Visual Studio Code
- Programming Language: Python
- Version Control : Github
- Natural Language Processing: NLTK
- Frameworks : PyTorch, Tensorflow
- Library: Hugging Face
- Data Visualization : Matplotlib



# Behavioral Model Result

Welcome to the Behavioral Round of the Technical Interview

Question 1:

Would you rather be liked or respected?

Your answer: I would rather be liked because liking by someone else helps me to learn more than getting respected.

Question 2:

How do you think this job will help you fulfill your career goals?

Your answer: I enjoy my job working in this position and I believe it will push me towards fulfilling my career goals.

Question 3:

What are your views about further studies?

Your answer: If I get opportunity to do research in computer science field and doing master in particular subject then I will definitely go for it..

Question 4:

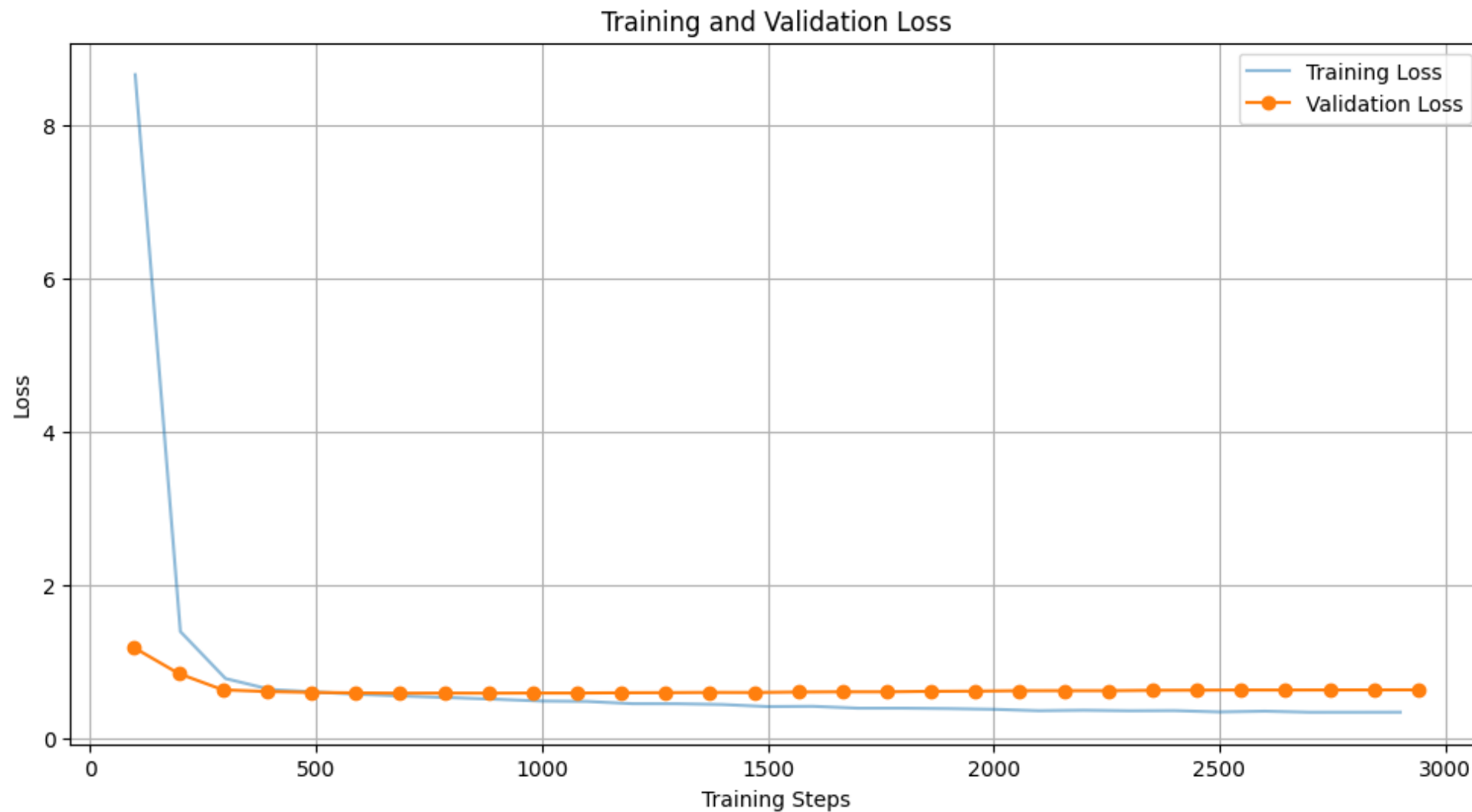
Tell us about a situation when your work has been criticized.

Your answer: I often get criticize when I tried to work alone rather than working on a team.

Question 5:

In what ways are you an organised or disorganised person?

# Behavioral Round Analysis



# Evaluation Metrics

Evaluation Metrics	Our Scores	Standard Scores
BLEU	77.26	<ul style="list-style-type: none"><li>• 0-10: Poor</li><li>• 10-30: Understandable</li><li>• 30-50: Medium</li><li>• 50-70: High Quality</li><li>• 70-100: Very High Quality</li></ul>
ROUGE-1	0.166	<ul style="list-style-type: none"><li>• <math>\geq 0.5</math>: Excellent</li><li>• <math>&gt; 0.5</math>: Good</li><li>• 0.4 - 0.5: Moderate</li></ul>
ROUGE-2	0.010	<ul style="list-style-type: none"><li>• <math>\geq 0.4</math>: Excellent</li><li>• <math>&gt; 0.4</math>: Good</li><li>• 0.2 - 0.4: Moderate</li></ul>
ROUGE-L	0.148	<ul style="list-style-type: none"><li>• <math>\geq 0.4</math>: Excellent</li><li>• <math>&gt; 0.4</math>: Good</li><li>• 0.2 - 0.4: Moderate</li></ul>

# Technical Model Results -[1]

--- Starting HTML section (Time limit: 5 minutes) ---



Time left for HTML: 4:59

Total time left: 44:59

Question (Easy): How can you generate a Question in HTML

Your answer: You can generate a question in HTML using the `<form>` element with `<label>` and `<input>` tags.

Time left for HTML: 4:39

Total time left: 44:39

Question (Easy): How can you create, use, and use tags from a simple `HTML form?`

Your answer: Create tags by defining `<input>` elements in a form, use them by submitting the form data, and handle tags in the `backend` or client-side script.

Time left for HTML: 1:49

Total time left: 41:49

Question (Easy): How do you create a custom HTML5 `backend in a Node.js` API?

Your answer: Create a custom HTML5 backend in Node.js by setting up a `server` with that serves HTML pages and handles form submissions.

Time left for HTML: 0:34

Total time left: 40:34

Question (Easy): Give an example of generating an HTML question about Forms that relates to form, `server`, handles from the `previous answer`.

Your answer: A server processes form data by parsing the submitted HTML form's input values and then handling them as specified in the Node.js backend code.

Progressing to Medium difficulty questions.

--- Starting CSS section (Time limit: 3 minutes) ---

Time left for CSS: 2:59

Total time left: 39:11

Question (Easy): How do you create a custom CSS framework for making a server?

# Technical Model Results -[2]

Question (Easy): Give an example of a Node.js file using File System.

Your answer: In a Node.js file, you can use the ``fs`` module to read a file by requiring the module, calling ``fs.readFile()`` with the file name and encoding, and handling the file cor

Time left for Node.js: 13:10

Total time left: 20:03

Question (Easy): Give an example of a `file fs` file from an Express.js application.

Your answer: In an Express.js application, you might use the ``fs`` module to serve static files. For example, you could use ``fs.readFile()`` to read and send a file's contents in `respo`

Progressing to Medium difficulty questions.

Time left for Node.js: 11:56

Total time left: 18:50

Question (Medium): Give an example of a URL that is used in Node.js `HTTP servers`.

Your answer: An example URL for a Node.js HTTP server could be ``http://localhost:3000``, where ``localhost`` refers to the local machine, and ``3000`` is the port number on which the serv

Time left for Node.js: 11:00

Total time left: 17:54

Question (Medium): How can you implement a module to enable voiceovers in a Node.js application?

Your answer: To enable voiceovers in a Node.js application, install a text-to-speech library like ``google-tts-api``, then create a function to convert text to speech and integrate it

Time left for Node.js: 9:47

Total time left: 16:41

Question (Medium): `Building on your previous answer` about generate, `voiceovers`, create, can you explain another Medium concept related to Event loop in Node.js?

Your answer: `**Microtasks**` in the Node.js event loop, which include operations from promises and ``process.nextTick()``, are executed before the next macro task, such as I/O operation

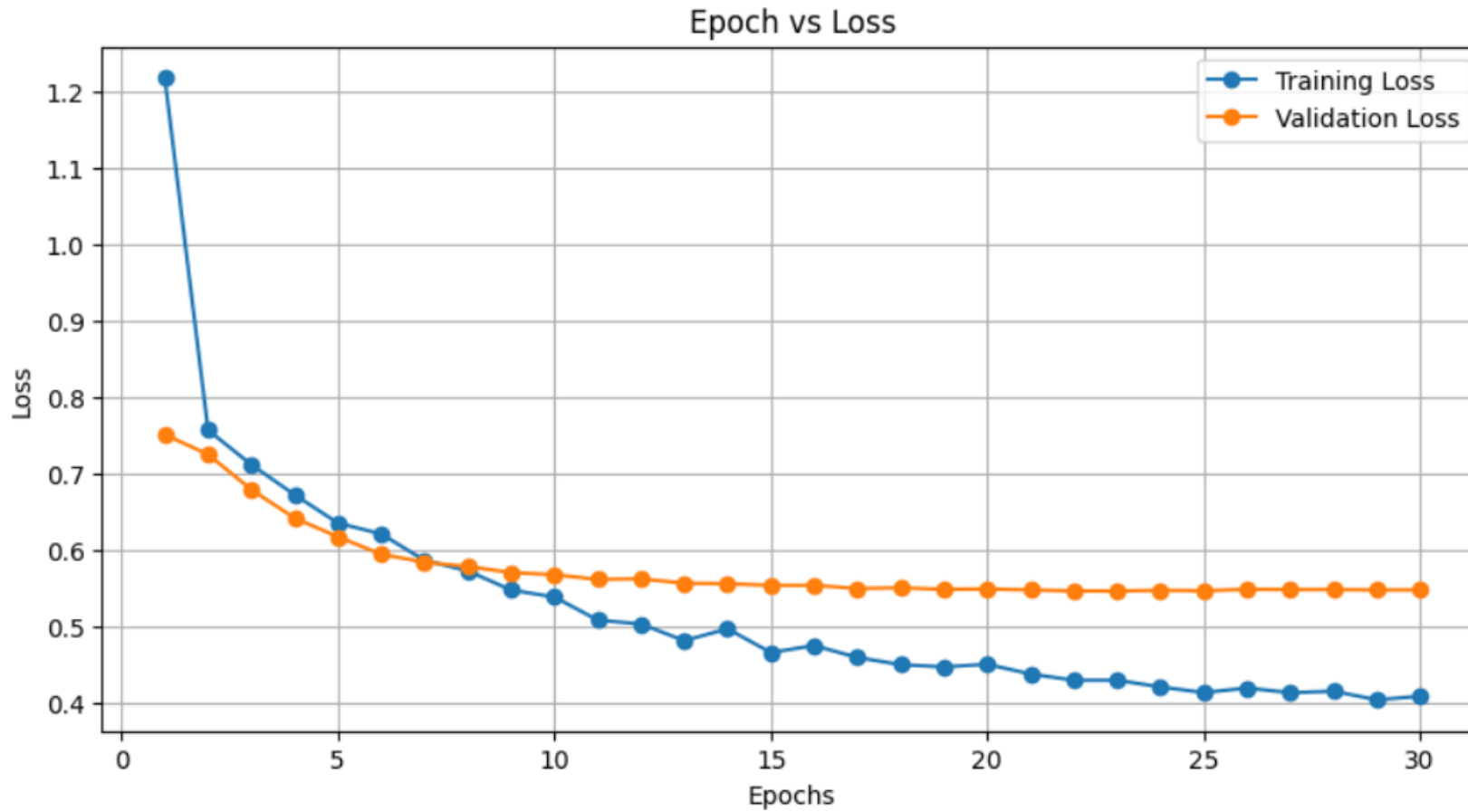
Time left for Node.js: 8:37

Total time left: 15:31

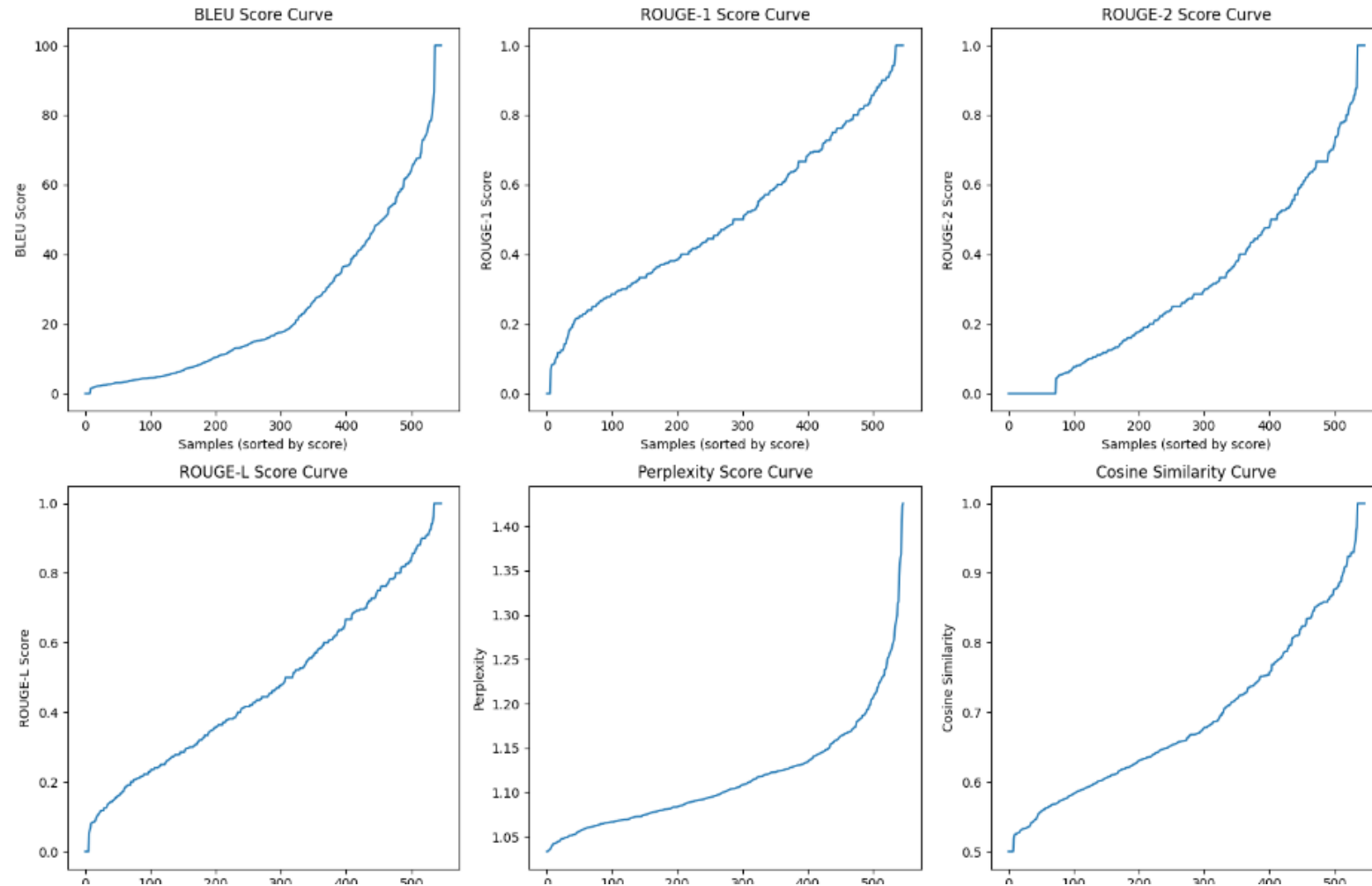
Question (Medium): Give an example of File system for Node.js with Files that corresponds to `nexttick`, next, microtasks.

Your answer: In Node.js, you can use ``fs.readFile()`` to read a file, then use ``process.nextTick()`` to schedule a callback for the next iteration of the event loop, and a Promise to f

# Technical Round Analysis -[1]



# Technical Round Analysis -[2]



# Evaluation Metrics -[1]

Evaluation Metrics	Our Scores	Standard Scores
BLEU	25.97	<ul style="list-style-type: none"><li>• 0-10: Poor</li><li>• 10-30: Understandable</li><li>• 30-50: Medium</li><li>• 50-70: High Quality</li><li>• 70-100: Very High Quality</li></ul>
ROUGE-1	0.50	<ul style="list-style-type: none"><li>• <math>\geq 0.5</math>: Excellent</li><li>• <math>&gt; 0.5</math>: Good</li><li>• 0.4 - 0.5: Moderate</li></ul>
ROUGE-2	0.32	<ul style="list-style-type: none"><li>• <math>\geq 0.4</math>: Excellent</li><li>• <math>&gt; 0.4</math>: Good</li><li>• 0.2 - 0.4: Moderate</li></ul>
ROUGE-L	0.47	<ul style="list-style-type: none"><li>• <math>\geq 0.4</math>: Excellent</li><li>• <math>&gt; 0.4</math>: Good</li><li>• 0.2 - 0.4: Moderate</li></ul>



# Evaluation Metrics -[2]

Perplexity	1.11	$\approx 1$ : Excellent
Cosine Similarity	0.69	<ul style="list-style-type: none"><li>• 1: Perfect Similarity</li><li>• 0.8 to 1: High Similarity</li><li>• 0.6 to 0.8: Moderate to High Similarity</li><li>• 0.4 to 0.6: Moderate Similarity</li><li>• 0.2 to 0.4: Low to Moderate Similarity</li><li>• 0 to 0.2: Low Similarity</li></ul>
METEOR	0.47	<ul style="list-style-type: none"><li>• 0: Poor Translation</li><li>• 1: Perfect Translation</li></ul>

# Remaining Tasks

- Increasing the question dataset.
- Using different fine tuning techniques and hyperparameter adjustment to increase the performance.
- Integrating the behavioral question model of with Technical question generation model of T5.
- Integrating TTS and SST with final model.
- Developing real time video interview interface.

# References – [1]

- [1] F. Chopra and I. Haalad, "Conducting Qualitative Interviews with AI(2023)," Cesifo Working Papers, p. 10666, 2023.
- [2] B. C. Lee and B. Y. Kim, "Development of an AI-based Interview System for Remote Hiring," International Journal of Advanced Research in Engineering and Technology (IJARET), vol. 12, no. 3, pp. 654-663, 2021.
- [3] A. Nigam, R. Pasricha, T. Singh and P. Churi, "A Systematic Review on AI-based Proctoring Systems: Past, Present and Future," Educ Inf Technol, vol. 26, pp. 6421-6445, 2021.
- [4] P. R. S B, M. Agnihotri and D. B. Jayagopi, "Improving Asynchronous Interview Interaction with Follow-up Question Generation," International Institute of Information Technology, vol. 6, pp. 79-89, 2021.

## References – [2]

- [5] T. R.K, B. Shinde, N. Rasal and S. Ghorpade, "Smart Interview System using AI Technology," International Research Journal of Modernization in Engineering Technology and Science, vol. 6, no. 2, 2024.
- [6] R. Mengi, H. Ghorpade and A. Kakade, "Fine-tuning T5 and RoBERTa Models for Enhanced Text Summarization and Sentimental Analysis," 2023.
- [7] N. Mulla and P. Gharpure, "Automatic question generation: a review of methodologies, datasets, evaluation metrics, and applications, Progress in Artificial Intelligence, vol. 12, pp. 1-32, 2023.

## References – [3]

- [8] M. Wng, P. Xle, Y. Du and X. Hu, "T5-Based Model for Abstractive Summarization: A Semi-Supervised Learning Approach with Consistency Loss Functions," Applied Sciences, vol. 13, no. 12, 2023.
- [9] V. D. Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior and K. Kavukcuoglu, "Wavenet: A generative model for raw audio," arXiv preprint arXiv:1609.03499, vol. 12, 2016.