EDI: EndSem Assessment SEM 2 2024

Vishwakarma Institute of Technology, Pune-37

EchoNotes

Your audio notes transformed

Project Guide: Dr.Radhika Kulkarni

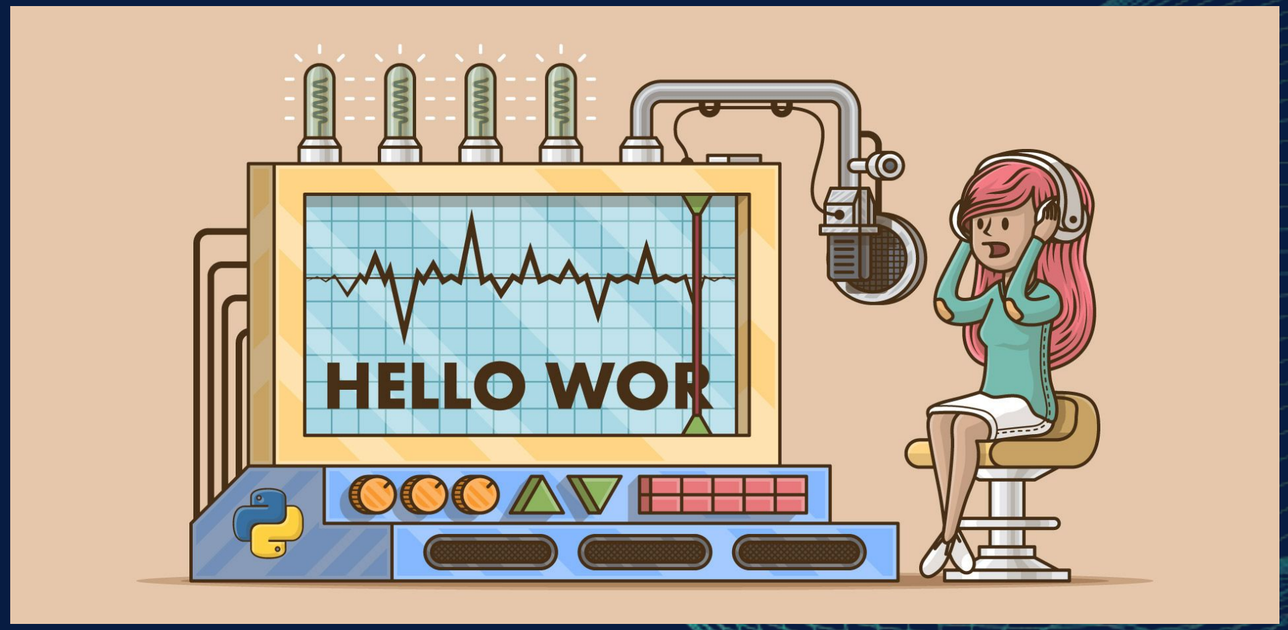
Group : TY-04

22 ~ 12111386 Avanish Shilimkar

28 ~ 12220306 Shreyash Bandi

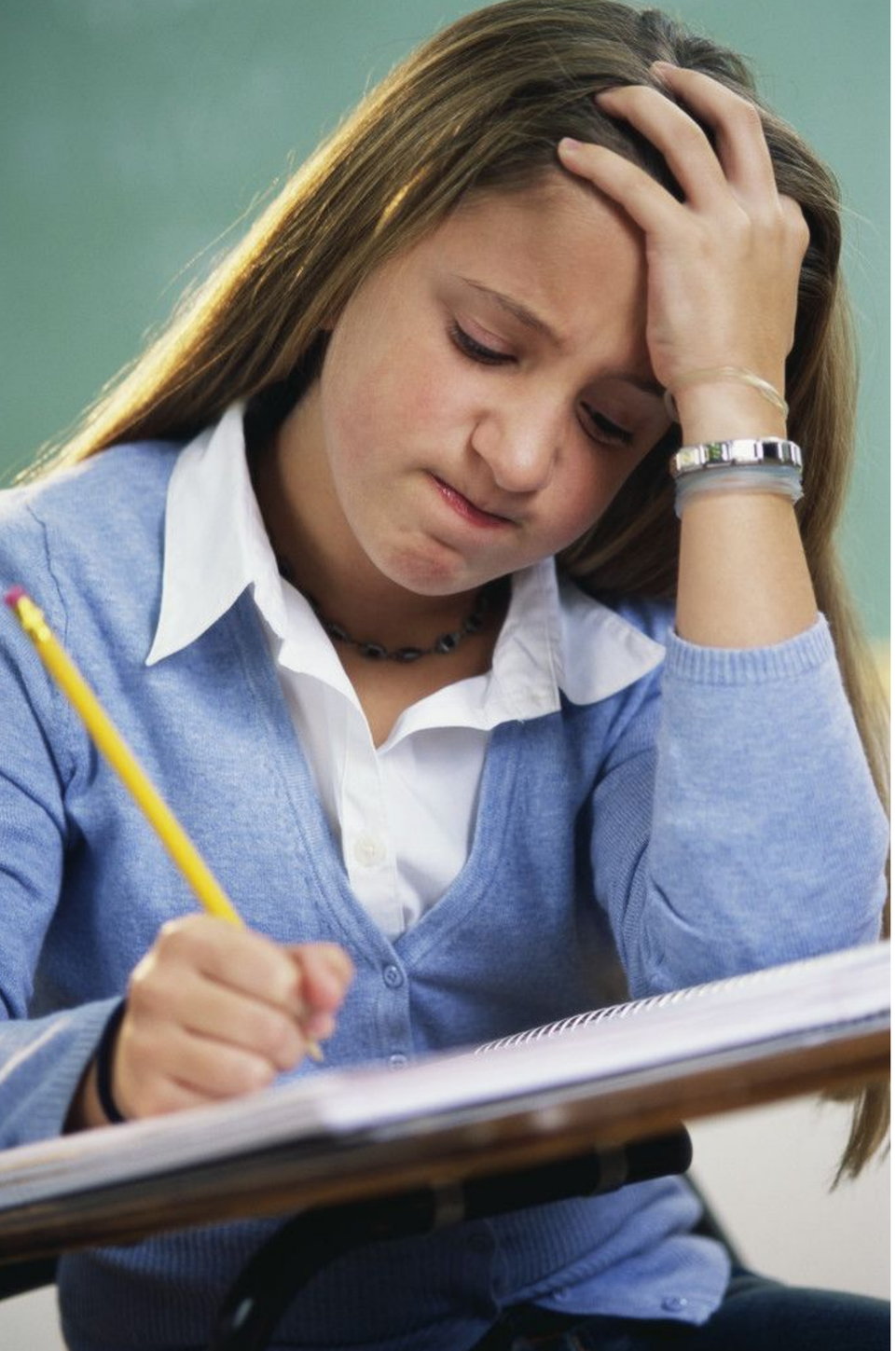
35 ~ 12111527 Sakshi Bhegade

64 ~ 12110169 Jidnyasa Dadmal



Agenda

1. Motivation
2. Problem Statement
3. Research gaps
4. Objectives
5. System architecture
6. Implementation
7. Conclusion
8. Future Plans



Motivation

The Challenge of Effective Note-Taking

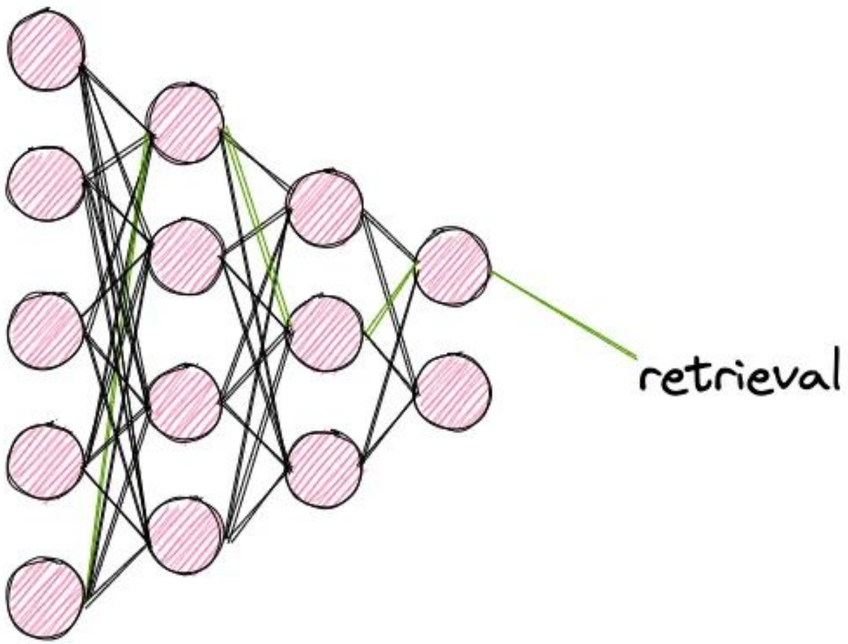
- Students often struggle to keep pace with the rapid flow of information in lectures.
- Traditional note-taking methods can be time-consuming and cumbersome.
- Difficulty in capturing key points and summarizing complex information.
- Inefficient note organization and retrieval can pose a challenge during revision.
- Provide a way to making studying fun.

Motivation

TEACHER-READY RESEARCH REVIEW

Practice Tests, Spaced Practice, and Successive Relearning: Tips for Classroom Use and for Guiding Students' Learning

John Dunlosky and Katherine A. Rawson
Kent State University



Learners retain approximately:

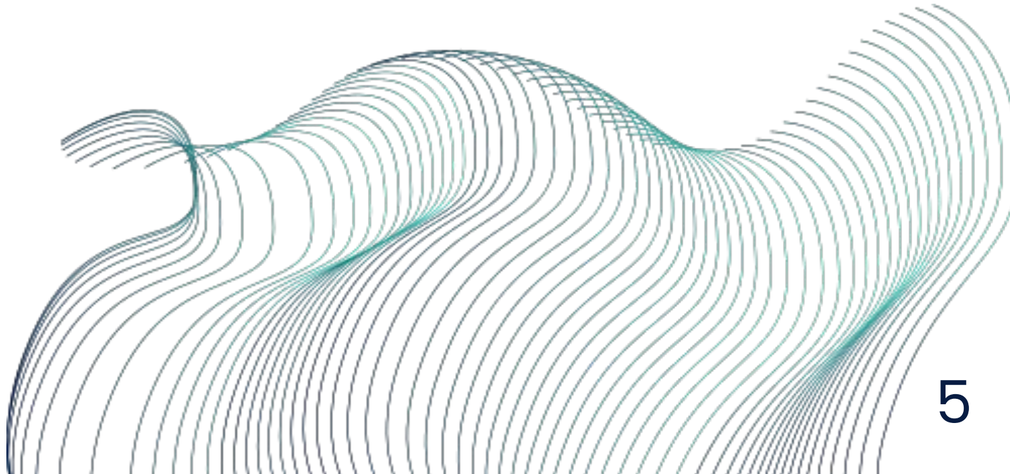
5% of what they learn when they've learned from lecture.

10% of what they learn when they've learned from reading.

20% of what they learn from audio-visual.

2

Problem Statement



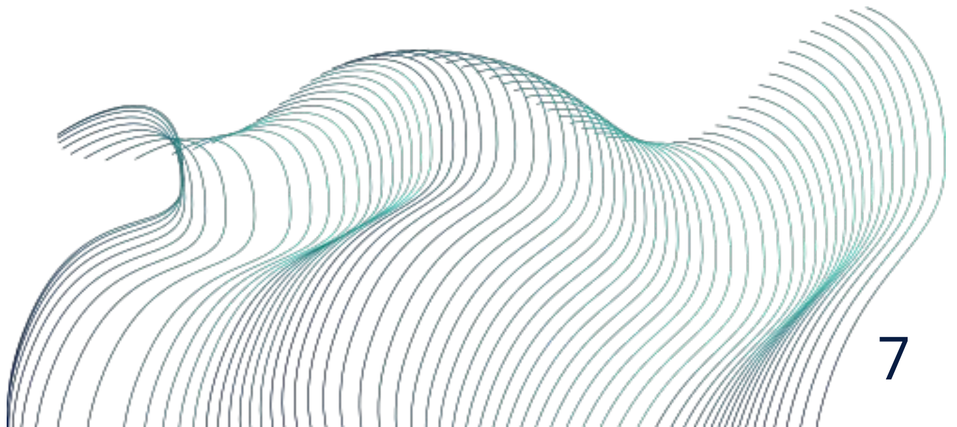
Problem Statement

- The current note-taking platforms have limited functionality, including its lack of diverse note-taking options and offline access, hinders efficient information capture, organization, and retrieval for users with varying learning styles and workflows.

- The high cost of premium note-taking apps with advanced features due to international pricing models is limiting accessibility for budget-conscious students and professionals.

3

Research Gaps



Research gaps

Multi-Module Support

- Current Research Handle text or audio separately. Our System: Summarizes audio, video, and text inputs.

Requirement for a intuitive interface

- Existing Systems require users to switch between multiple tools to accomplish related tasks.

Language Support

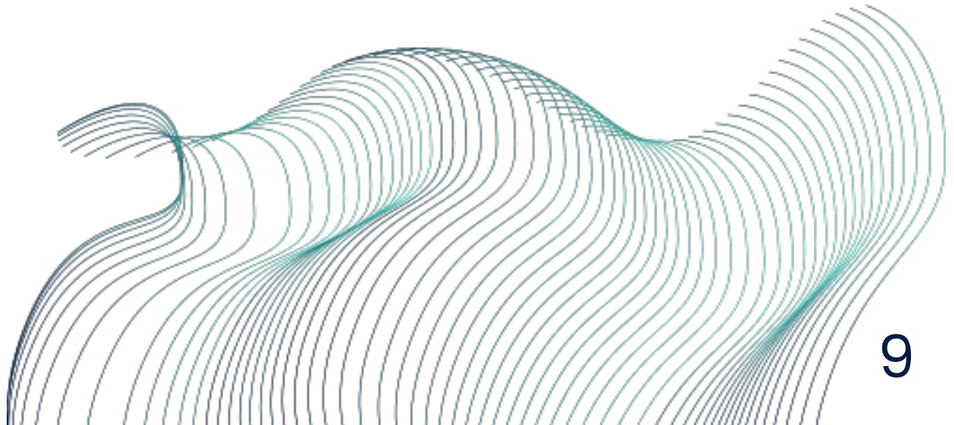
- Current Systems offer limited language support, focusing primarily on English and neglecting other languages.

Accuracy and Efficiency

- Some systems prioritize either speed or accuracy in summarization tasks.

4

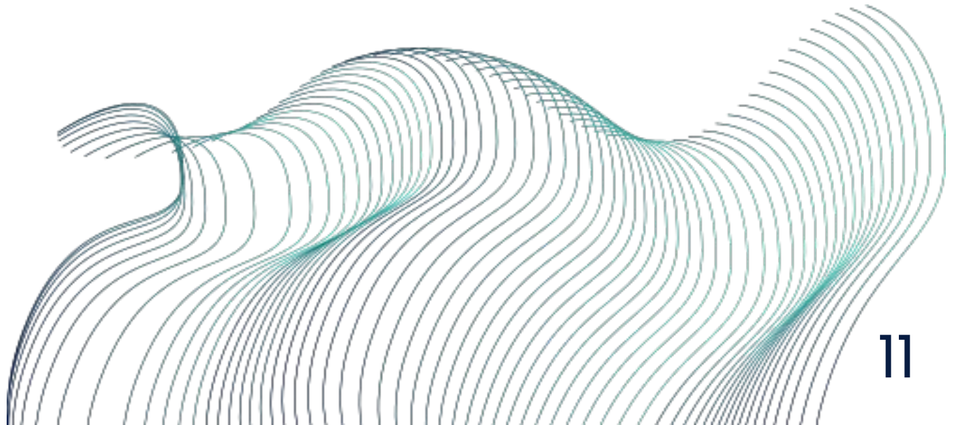
Objectives



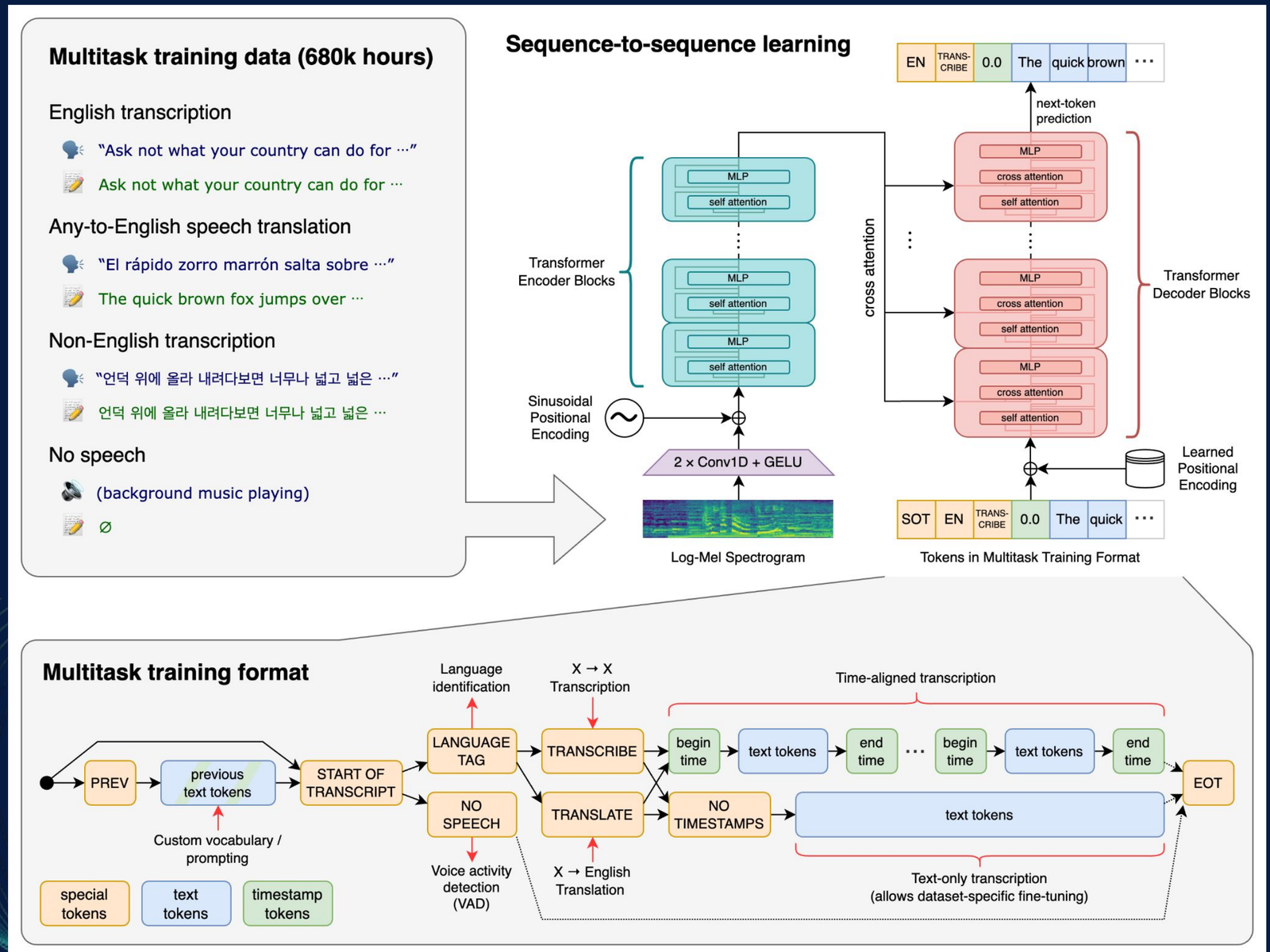
Develop an Integrated Web Application AI Note-Taking System:

- **Generate Flashcards for Efficient Learning**
- **Creating Mind Maps**
- **Summarization**
- **Formatting Notes**
- **Messages/Creating Group Chats for Sharing Notes**
- **Exporting the Notes in Various Formats like .txt, .pdf, .doc**
- **Provide Quizzes in spaced repetition format.**

5 METHODOLOGY



Whisper - OpenAI model



Dataset size	English WER (↓)	Multilingual WER (↓)	X→En BLEU (↑)
3405	30.5	92.4	0.2
6811	19.6	72.7	1.7
13621	14.4	56.6	7.9
27243	12.3	45.0	13.9
54486	10.9	36.4	19.2
681070	**9.9**	**29.2**	**24.8**

Transformers are composed of two main parts: the encoder and the decoder. Each of these parts is made up of multiple layers, and each layer contains two crucial elements: the multi-head self-attention mechanism and a feed-forward neural network.

The self-attention mechanism is the heart of the transformer. It allows the model to weigh the importance of each word in a sentence relative to all the other words. This means the model can understand the context in which a word appears. For example, in the sentence 'The cat sat on the mat,' the word 'cat' has a relationship with 'sat' and 'mat.' Self-attention helps the model grasp these relationships.

Whisper - OpenAI model

Embedding Words into Vectors:
First, each word in the sentence is converted into an embedding vector. Let's assume we have simple embeddings like this (for illustration purposes, using 2D vectors):

```
"The" -> [1, 0]
"cat" -> [0, 1]
"sat" -> [1, 1]
"on" -> [0, 0]
"the" -> [1, 0]
"mat" -> [0, 1]
```

Whisper - OpenAI model

Creating Queries, Keys, and Values:
Each word vector is transformed into three vectors: Query (Q), Key (K), and Value (V). This is done using learned matrices (weights). For simplicity, let's use hypothetical transformations. Suppose our transformation matrices (Wq, Wk, Wv) are such that:

```
Q = Wq * word_vector
K = Wk * word_vector
V = Wv * word_vector
```

```
"sat" -> Q: [1, 1], K: [1, 0], V: [1, 1]
"cat" -> Q: [0, 1], K: [0, 1], V: [0, 1]
"the" -> Q: [1, 0], K: [1, 0], V: [1, 0]
"on" -> Q: [0, 0], K: [0, 0], V: [0, 0]
"mat" -> Q: [0, 1], K: [0, 1], V: [0, 1]
```

Whisper – OpenAI model

Calculating Attention Scores:

For each word, we calculate attention scores by taking the dot product of the Query of the word "sat" with the Keys of all words in the sentence:

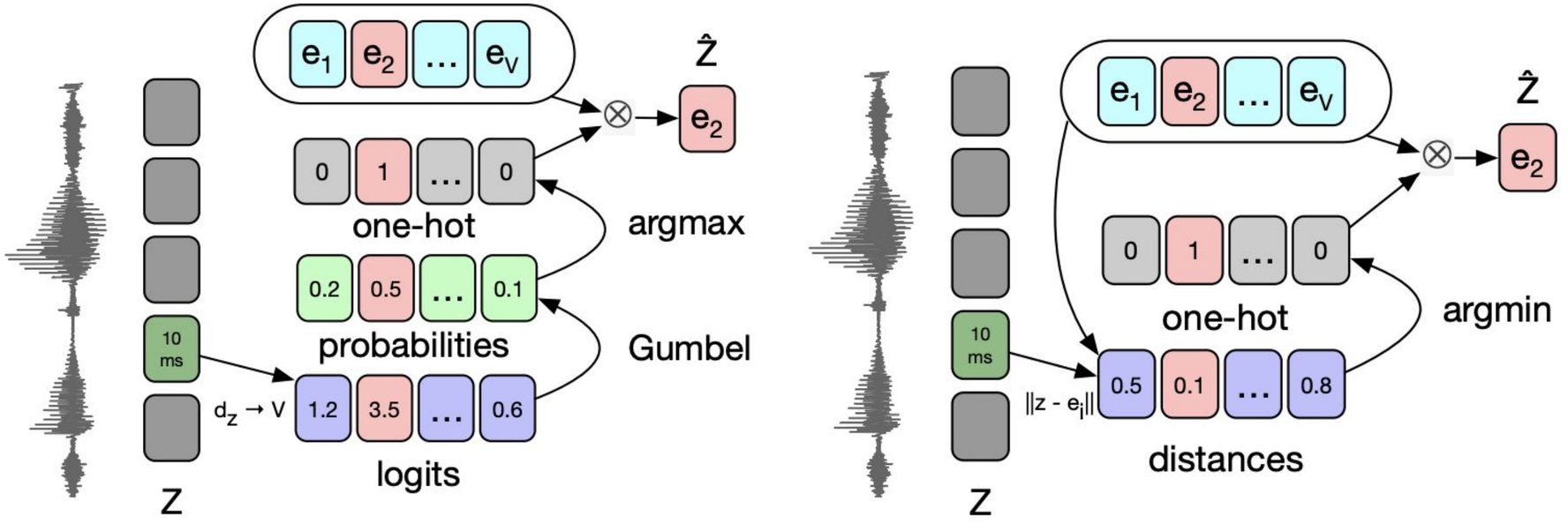
```
Attention score for "sat" with "the": dot([1, 1], [1, 0]) = 1*1 + 1*0 = 1
Attention score for "sat" with "cat": dot([1, 1], [0, 1]) = 1*0 + 1*1 = 1
Attention score for "sat" with "sat": dot([1, 1], [1, 0]) = 1*1 + 1*0 = 1
Attention score for "sat" with "on": dot([1, 1], [0, 0]) = 1*0 + 1*0 = 0
Attention score for "sat" with "mat": dot([1, 1], [0, 1]) = 1*0 + 1*1 = 1
```

Whisper – OpenAI model

Applying the Softmax Function:
Next, we apply the softmax function to these scores to normalize them, ensuring they sum to 1. This gives us the attention weights:

```
Attention score for "sat" with "the": dot([1, 1], [1, 0]) = 1*1 + 1*0 = 1
Attention score for "sat" with "cat": dot([1, 1], [0, 1]) = 1*0 + 1*1 = 1
Attention score for "sat" with "sat": dot([1, 1], [1, 0]) = 1*1 + 1*0 = 1
Attention score for "sat" with "on": dot([1, 1], [0, 0]) = 1*0 + 1*0 = 0
Attention score for "sat" with "mat": dot([1, 1], [0, 1]) = 1*0 + 1*1 = 1
```

Softmax is a mathematical function that converts a list of numbers into a list of probabilities. The probabilities will add up to 100%, or 1 when expressed as decimals.



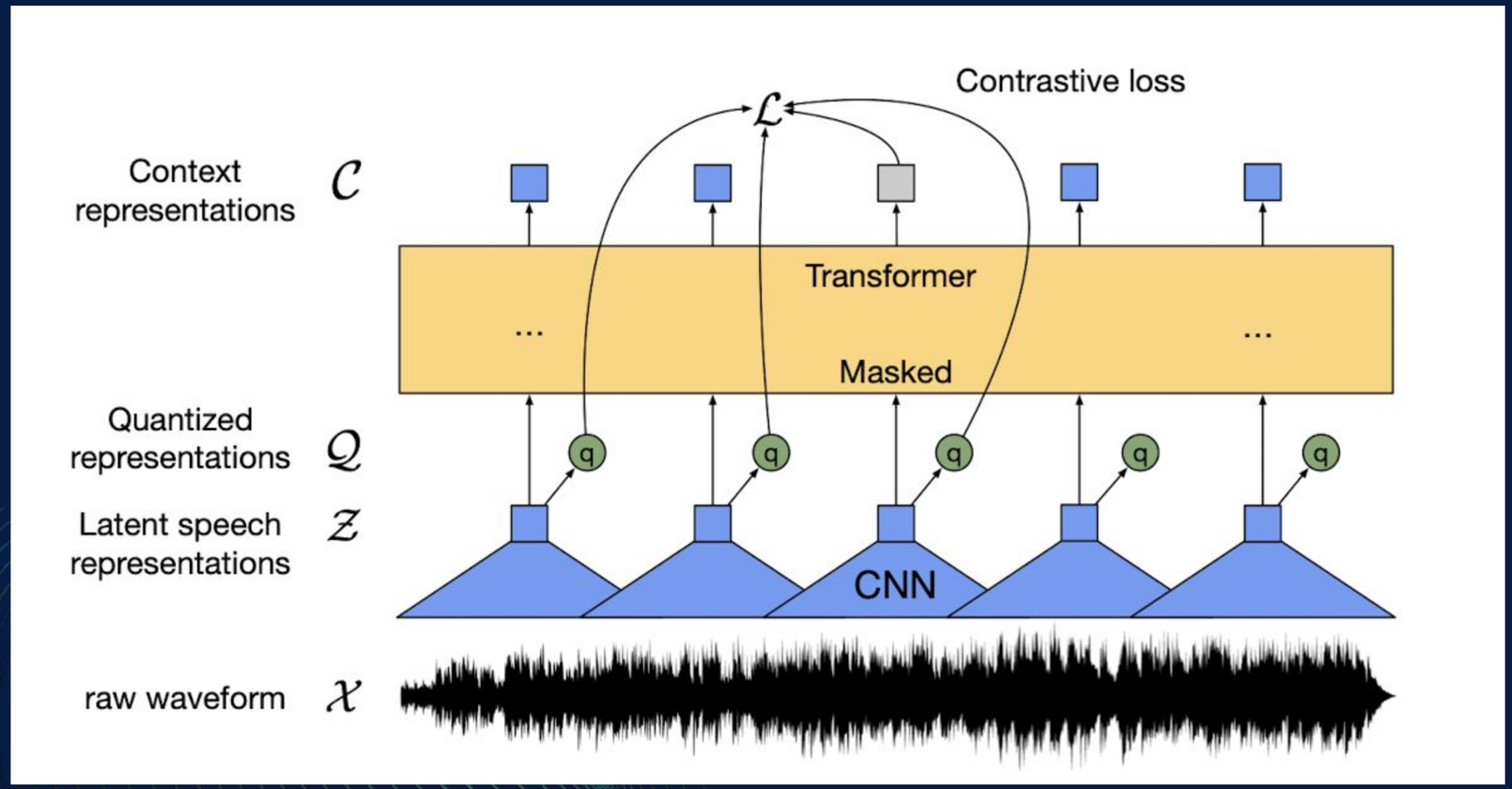
(a) Gumbel-Softmax

(b) K-means clustering.

Figure 2: (a) The Gumbel-Softmax quantization computes logits representing the codebook vectors ($\mathbf{e}$). In the forward pass the argmax codeword ($\mathbf{e}_2$) is chosen and for backward (not shown) the exact probabilities are used. (b) K-means vector quantization computes the distance to all codeword vector and chooses the closest (argmin).

How does Audio to Text work

wav2vec



wav2vec works by first converting raw audio signals into a series of detailed features using a neural network. It learns patterns from large amounts of unlabeled audio data, predicting missing parts of the audio to understand speech better. For example, if you have a recording of someone saying "hello," wav2vec analyzes the sound waves, extracts important speech characteristics, and builds a detailed understanding of the word "hello" even if parts of the audio are noisy or missing. This pre-trained model is then fine-tuned with labeled audio-to-text data to accurately

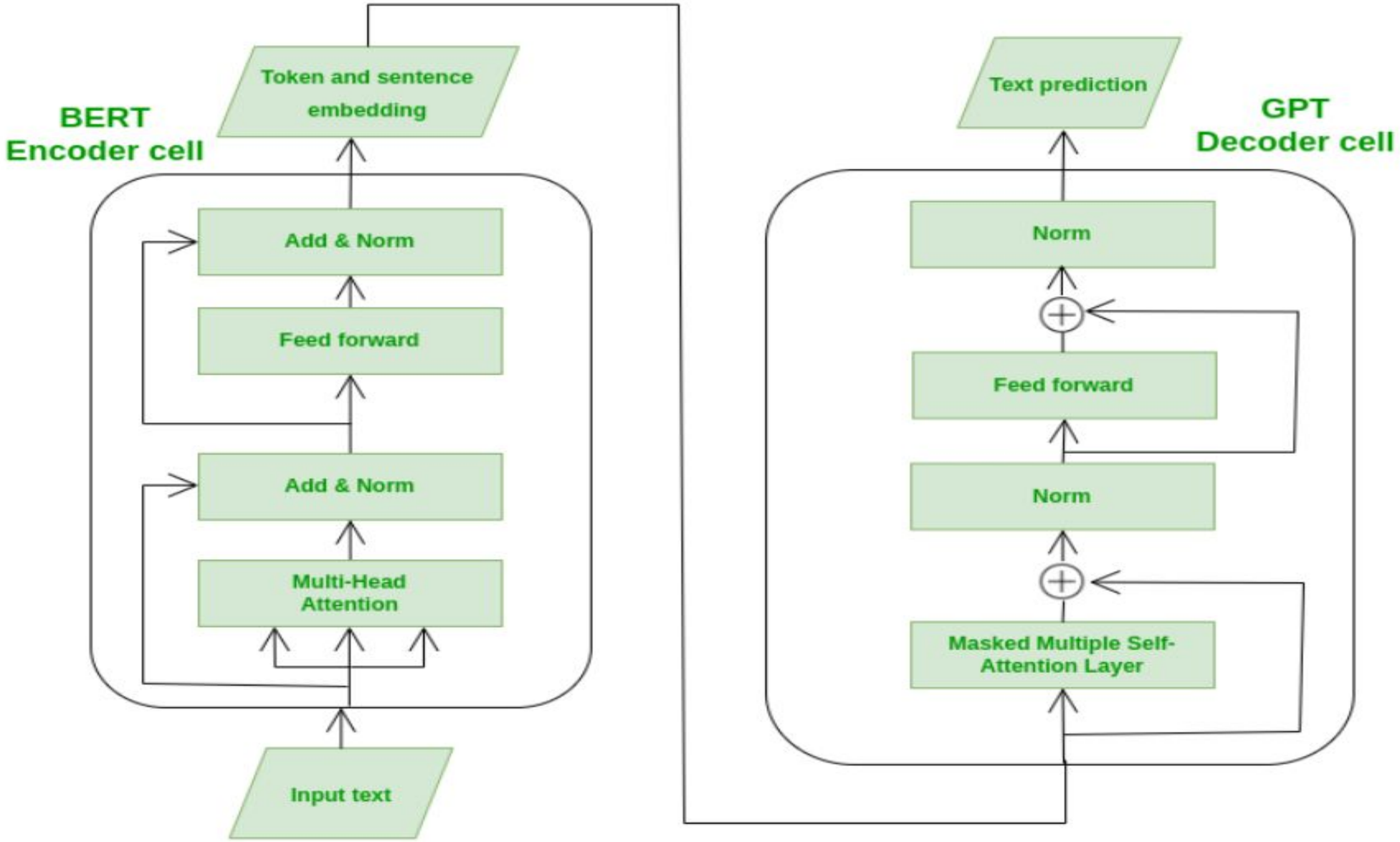
BART (Bidirectional and Auto-Regressive Transformers):

A sequence-to-sequence (seq2seq) model that uses a transformer architecture. It is pre-trained to reconstruct corrupted input sequences.

Key Components:

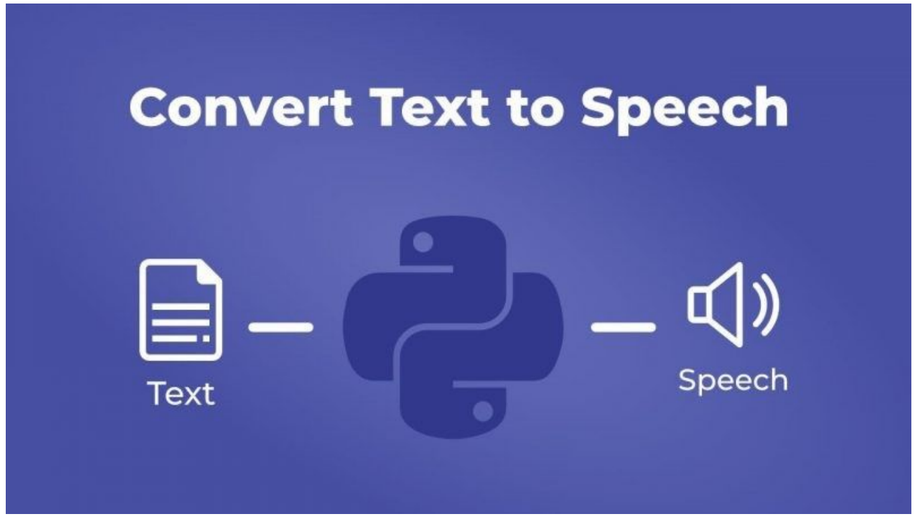
Encoder:
Reads the entire input text bidirectionally, similar to BERT.
Understands the context of each word in relation to others.

Decoder:
Generates the output text autoregressively, similar to GPT.
Predicts one word at a time based on previously generated words.

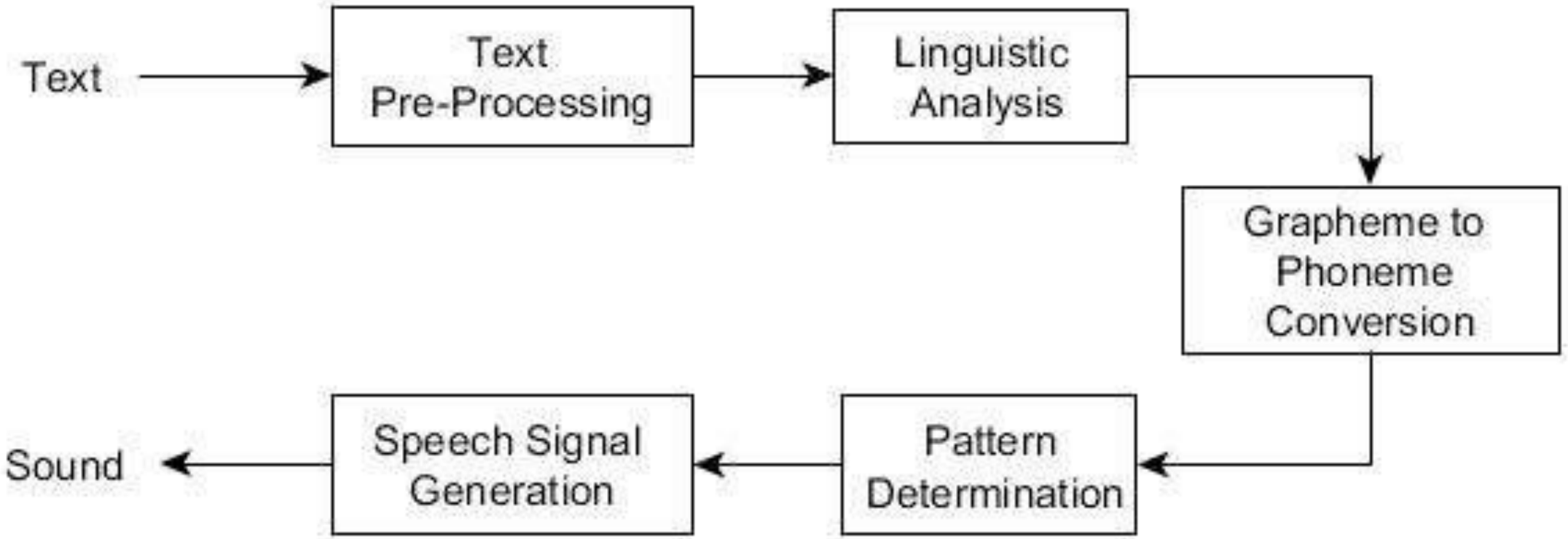

BERT
Encoder cell
Token and sentence embedding
Add & Norm
Feed forward
Add & Norm
Multi-Head Attention
Input text
Text prediction
GPT
Decoder cell
Norm
Feed forward
Norm
Masked Multiple Self-Attention Layer

gTTS (Google Text-to-Speech):

- A Python library and command-line interface (CLI) tool that interfaces with Google’s Text-to-Speech API.
- It enables conversion of text into spoken audio.
- Supports multiple languages and accents, providing high-quality audio output.



GTTS Working for Text-to-Speech Conversion



How YOUTUBE SUMMARIZATION WORKS ?

```
!pip install -q transformers
```

```
2.9 MB 2.5 MB/s
3.3 MB 32.4 MB/s
52 kB 1.3 MB/s
636 kB 69.3 MB/s
895 kB 67.4 MB/s
```

```
[ ]  !pip install -q youtube_transcript_api
```

```
[ ]  from transformers import pipeline
     from youtube_transcript_api import YouTubeTranscriptApi
```

```
[ ]  youtube_video = "https://www.youtube.com/watch?v=A4OmtyaB
```

```
[ ]  video_id = youtube_video.split("=")[1]
```

```
[ ]  video_id
```

How YOUTUBE SUMMARIZATION WORKS ?

```
[ ]  YouTubeTranscriptApi.get_transcript(video_id)
     transcript = YouTubeTranscriptApi.get_transcript(video_id)
```

```
[ ]  transcript[0:5]
```

```
[{'duration': 4.96,
  'start': 1.52,
  'text': "for germany it's the end of an era"},
 {'duration': 5.279,
  'start': 4.4,
  'text': "and as europe's biggest economy there"},
 {'duration': 4.72, 'start': 6.48, 'text': 'are some huge challenges ahead'},
 {'duration': 3.681, 'start': 9.679, 'text': 'from its increasingly complex'},
 {'duration': 3.12,
  'start': 11.2,
  'text': 'relationship with china to climate'}]
```

```
[ ]
     result = ""
     for i in transcript:
         result += ' ' + i['text']
     #print(result)
     print(len(result))
```

```
9081
```

How YOUTUBE SUMMARIZATION WORKS ?

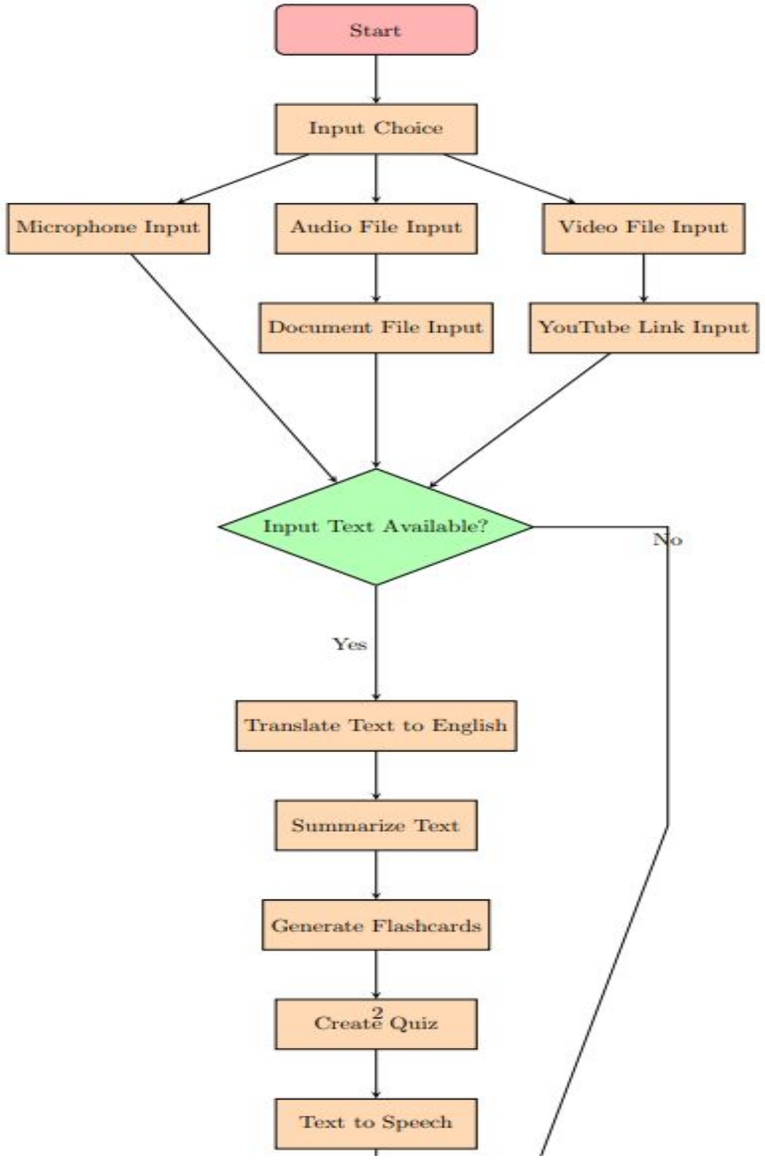
```
summarizer = pipeline('summarization')
```

```
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6
Downloading: 100% 1.76k/1.76k [
Downloading: 100% 1.14G/1.14G
Downloading: 100% 26.0/26.0 [00
Downloading: 100% 878k/878k [0
Downloading: 100% 446k/446k [0
```

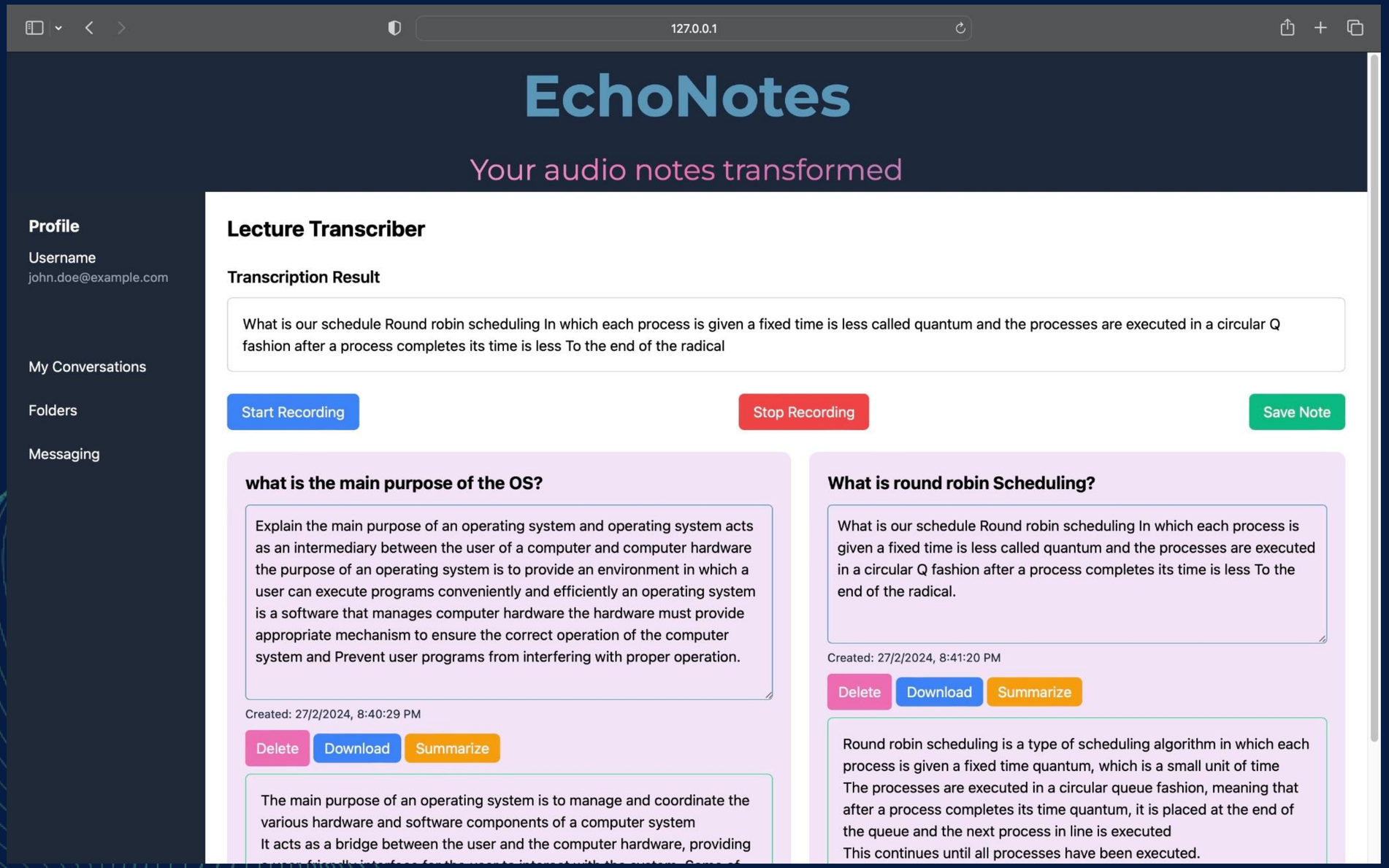
```
[ ]  num_iters = int(len(result)/1000)
     summarized_text = []
     for i in range(0, num_iters + 1):
       start = 0
       start = i * 1000
       end = (i + 1) * 1000
       print("input text \n" + result[start:end])
       out = summarizer(result[start:end])
       out = out[0]
       out = out['summary_text']
       print("Summarized text\n"+out)
       summarized_text.append(out)

     #print(summarized_text)
```

6

FLOWCHART

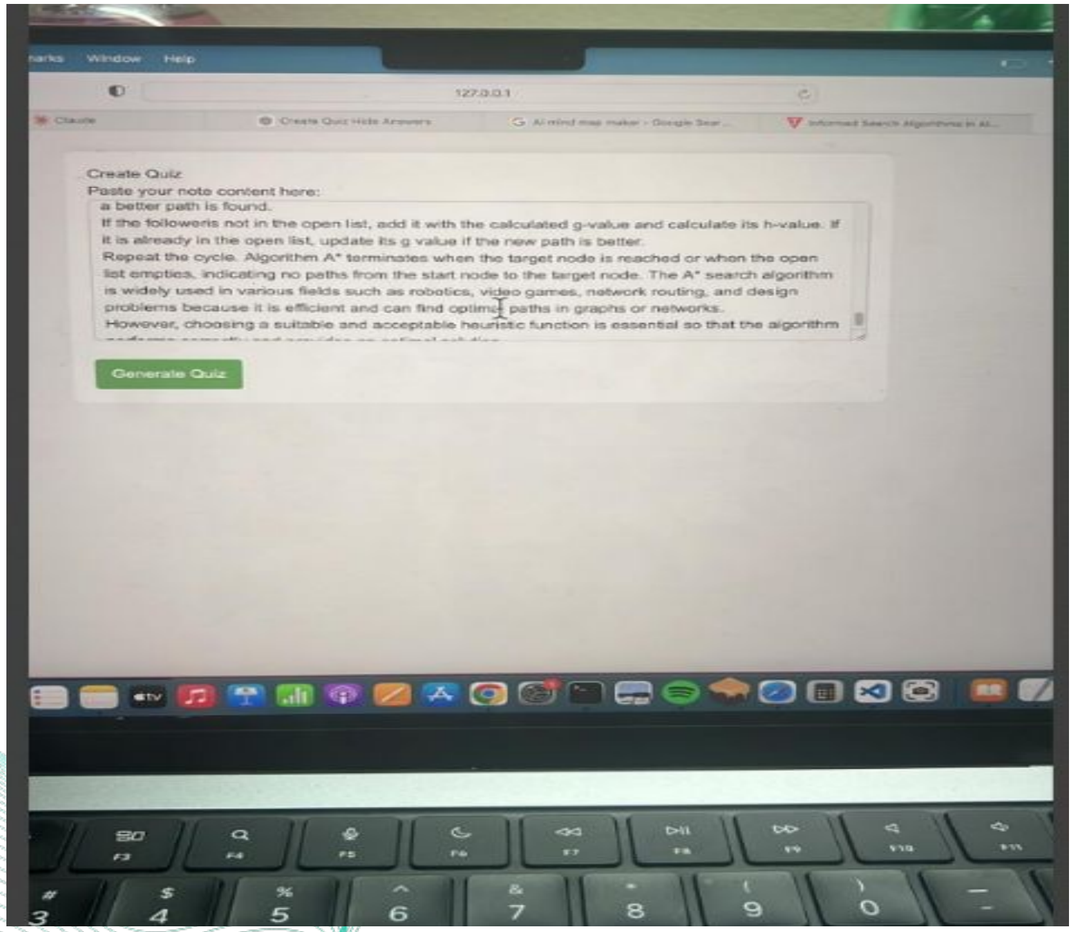

Start
Input Choice
Microphone Input
Audio File Input
Video File Input
Document File Input
YouTube Link Input
Input Text Available?
No
Yes
Translate Text to English
Summarize Text
Generate Flashcards
Create Quiz
Text to Speech

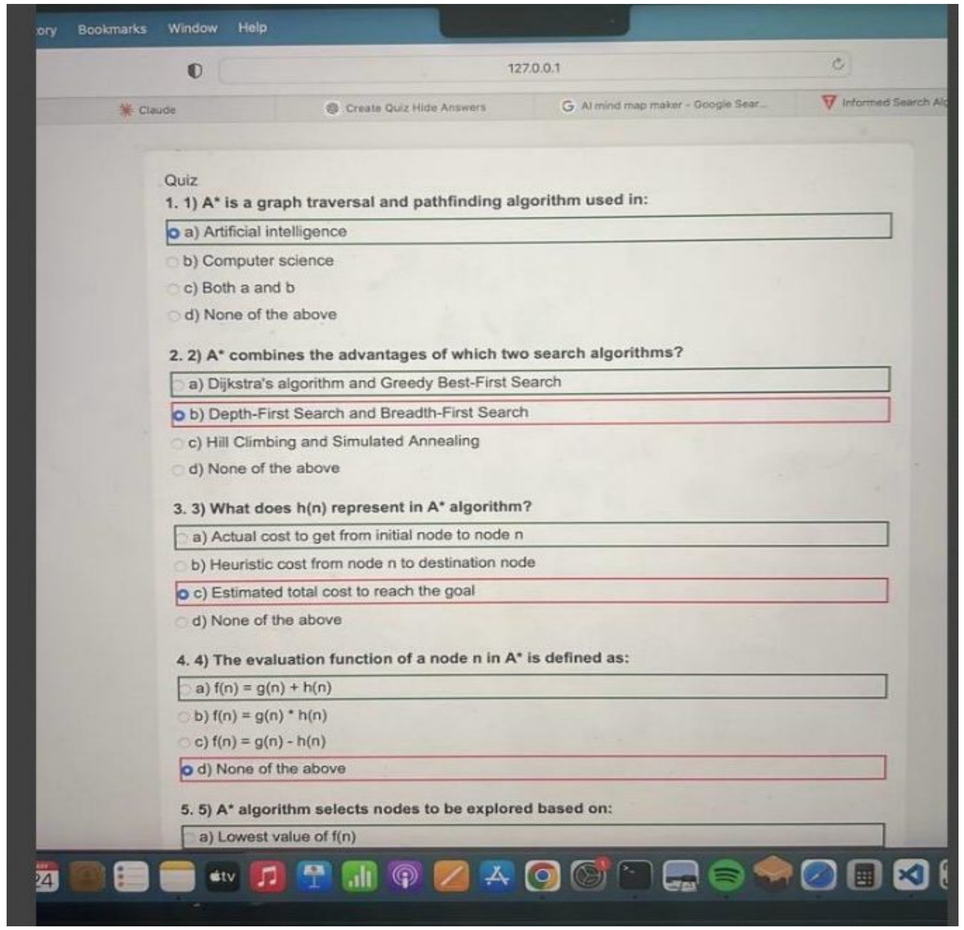
RESULT


127.0.0.1
EchoNotes
Your audio notes transformed
Profile
Username
john.doe@example.com
My Conversations
Folders
Messaging
Lecture Transcriber
Transcription Result
What is our schedule Round robin scheduling In which each process is given a fixed time is less called quantum and the processes are executed in a circular Q fashion after a process completes its time is less To the end of the radical
Start Recording
Stop Recording
Save Note
what is the main purpose of the OS?
Explain the main purpose of an operating system and operating system acts as an intermediary between the user of a computer and computer hardware the purpose of an operating system is to provide an environment in which a user can execute programs conveniently and efficiently an operating system is a software that manages computer hardware the hardware must provide appropriate mechanism to ensure the correct operation of the computer system and Prevent user programs from interfering with proper operation.
Created: 27/2/2024, 8:40:29 PM
Delete
Download
Summarize
The main purpose of an operating system is to manage and coordinate the various hardware and software components of a computer system
It acts as a bridge between the user and the computer hardware, providing
What is round robin Scheduling?
What is our schedule Round robin scheduling In which each process is given a fixed time is less called quantum and the processes are executed in a circular Q fashion after a process completes its time is less To the end of the radical.
Created: 27/2/2024, 8:41:20 PM
Delete
Download
Summarize
Round robin scheduling is a type of scheduling algorithm in which each process is given a fixed time quantum, which is a small unit of time
The processes are executed in a circular queue fashion, meaning that after a process completes its time quantum, it is placed at the end of the queue and the next process in line is executed
This continues until all processes have been executed.

IMPLEMENTATION

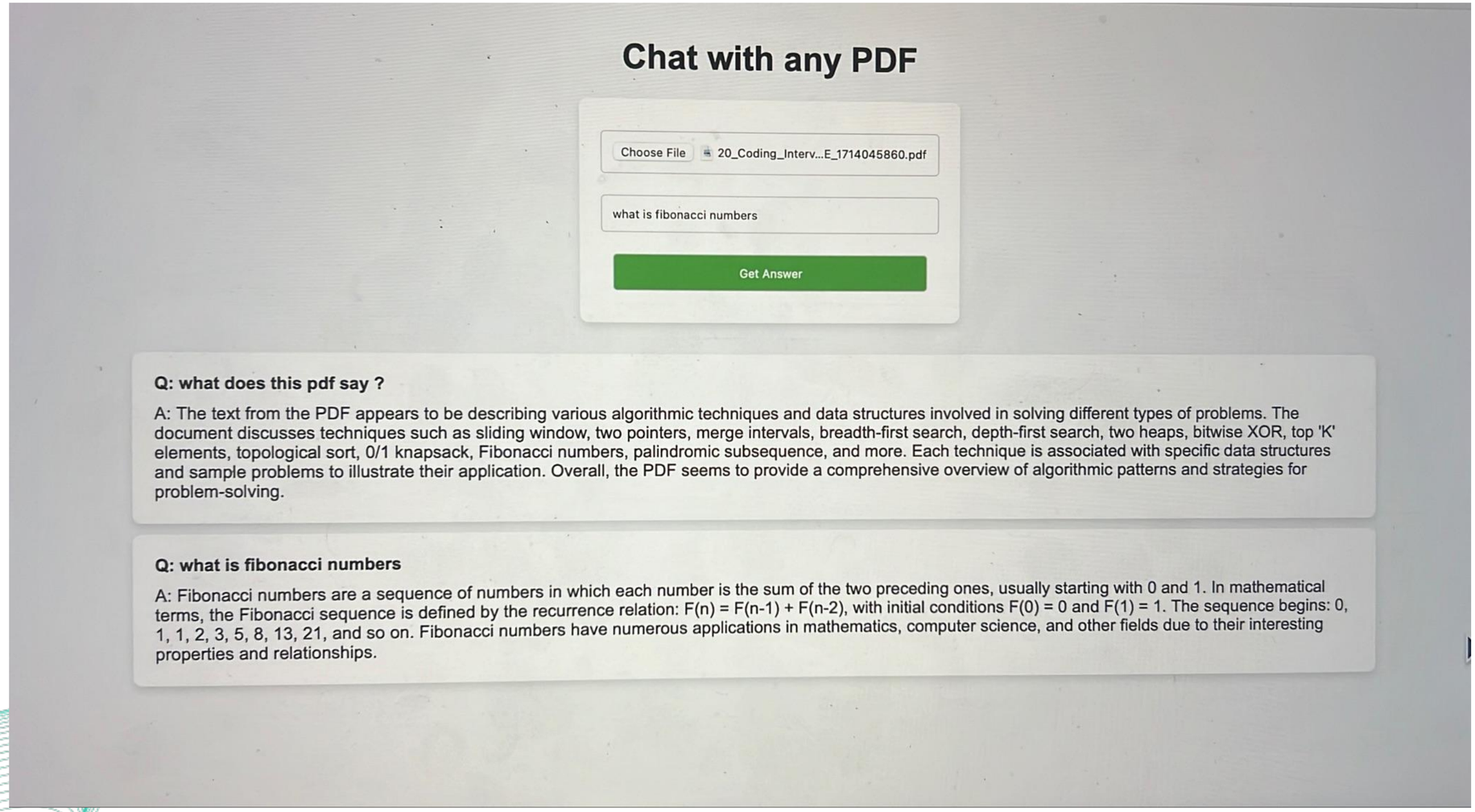
QUIZ GENERATION

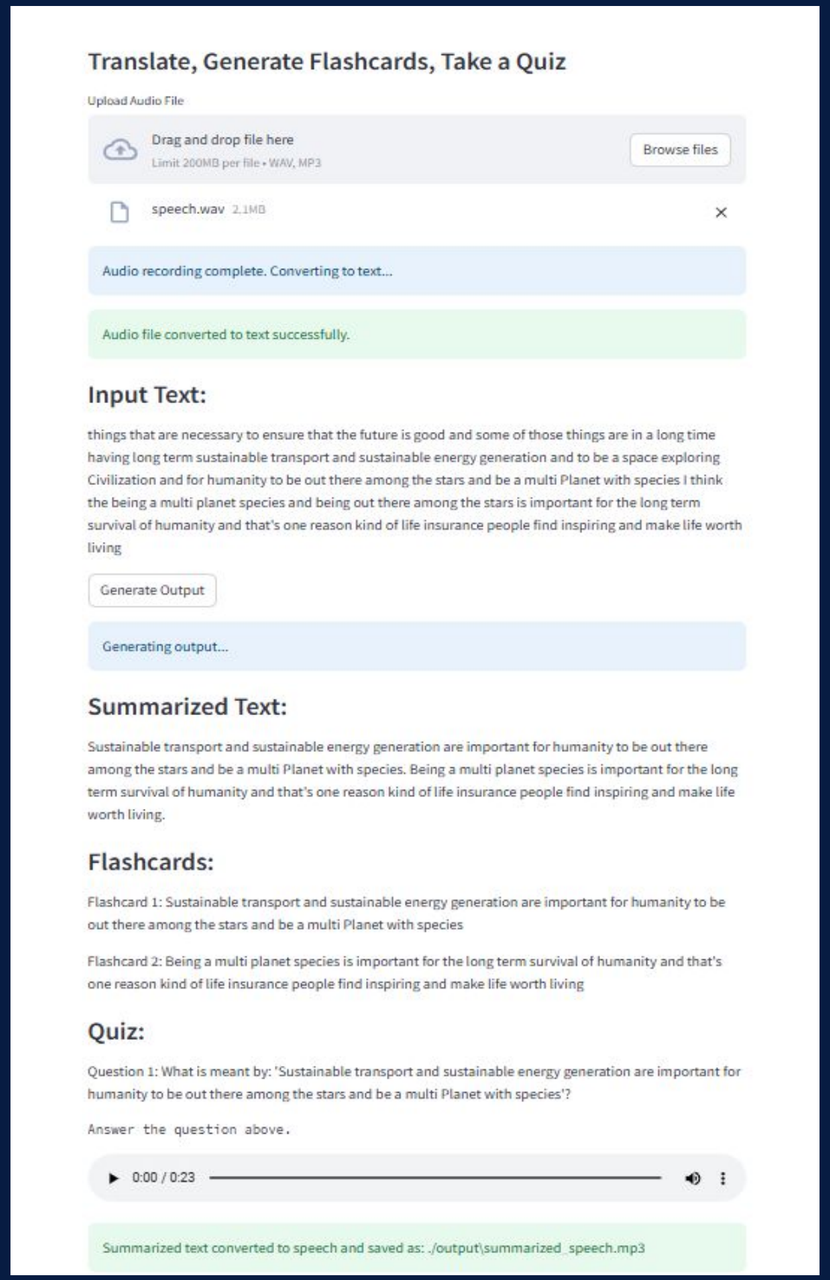


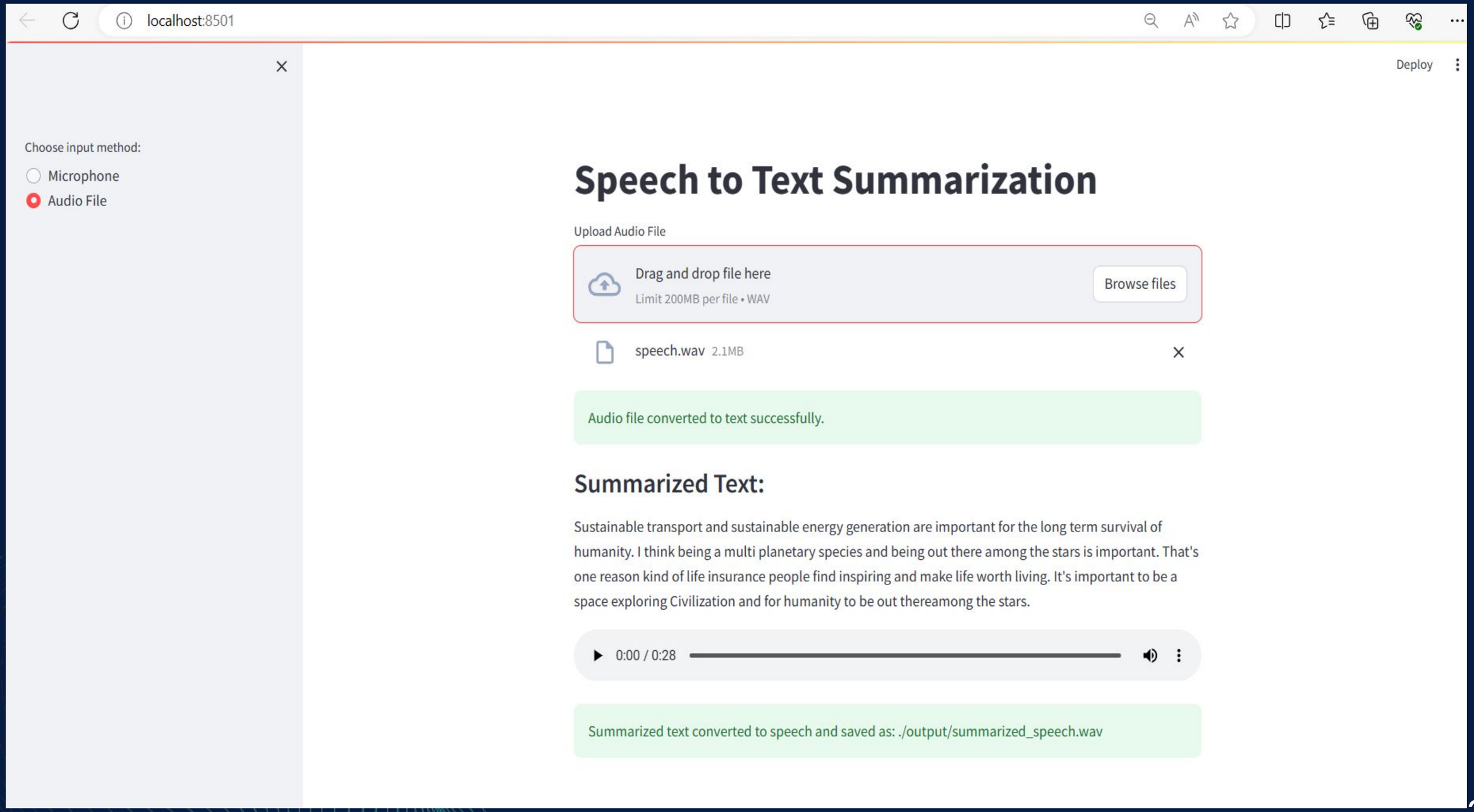


IMPLEMENTATION

CHAT WITH PDF

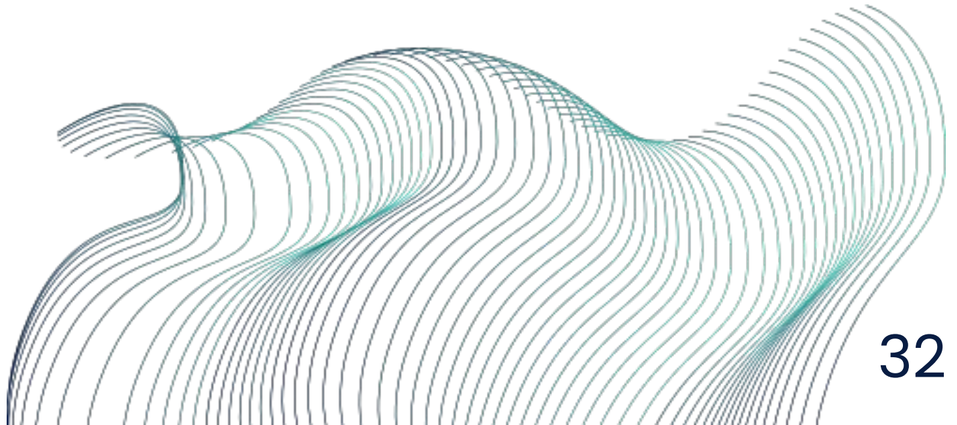



Translate, Generate Flashcards, Take a Quiz
Upload Audio File
Drag and drop file here
Limit 200MB per file • WAV, MP3
Browse files
speech.wav 2.1MB
Audio recording complete. Converting to text...
Audio file converted to text successfully.
Input Text:
things that are necessary to ensure that the future is good and some of those things are in a long time having long term sustainable transport and sustainable energy generation and to be a space exploring Civilization and for humanity to be out there among the stars and be a multi Planet with species I think the being a multi planet species and being out there among the stars is important for the long term survival of humanity and that's one reason kind of life insurance people find inspiring and make life worth living
Generate Output
Generating output...
Summarized Text:
Sustainable transport and sustainable energy generation are important for humanity to be out there among the stars and be a multi Planet with species. Being a multi planet species is important for the long term survival of humanity and that's one reason kind of life insurance people find inspiring and make life worth living.
Flashcards:
Flashcard 1: Sustainable transport and sustainable energy generation are important for humanity to be out there among the stars and be a multi Planet with species
Flashcard 2: Being a multi planet species is important for the long term survival of humanity and that's one reason kind of life insurance people find inspiring and make life worth living
Quiz:
Question 1: What is meant by: 'Sustainable transport and sustainable energy generation are important for humanity to be out there among the stars and be a multi Planet with species'?
Answer the question above.
0:00 / 0:23
Summarized text converted to speech and saved as: ./output\summarized_speech.mp3


localhost:8501
Deploy
Choose input method:
Microphone
Audio File
Speech to Text Summarization
Upload Audio File
Drag and drop file here
Limit 200MB per file • WAV
Browse files
speech.wav 2.1MB
Audio file converted to text successfully.
Summarized Text:
Sustainable transport and sustainable energy generation are important for the long term survival of humanity. I think being a multi planetary species and being out there among the stars is important. That's one reason kind of life insurance people find inspiring and make life worth living. It's important to be a space exploring Civilization and for humanity to be out thereamong the stars.
0:00 / 0:28
Summarized text converted to speech and saved as: ./output/summarized_speech.wav

7

Future Plan



Future Plan

- **Real-Time Processing:**

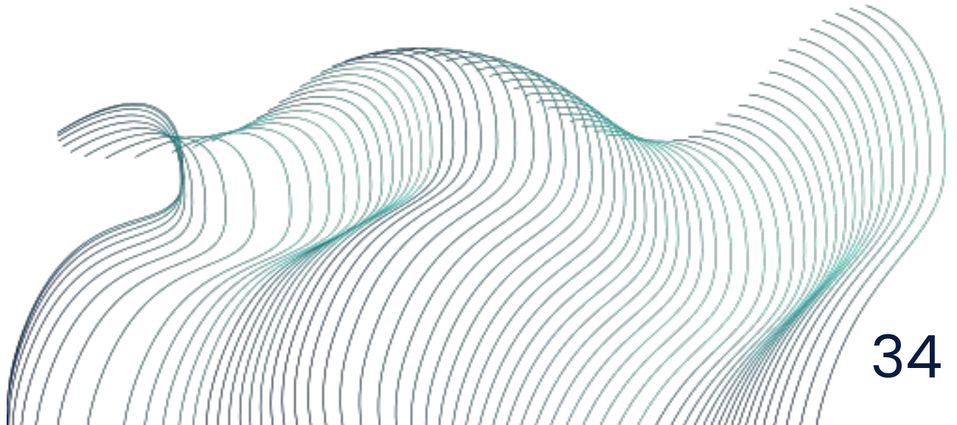
Live Transcription and Translation: Integrate real-time speech-to-text and translation features for live events, meetings, and webinars.

- Providing accurate mind-maps and suggestions for the user to help better understanding.

- To provide the interface as an Application rather than on web to allow faster processing.

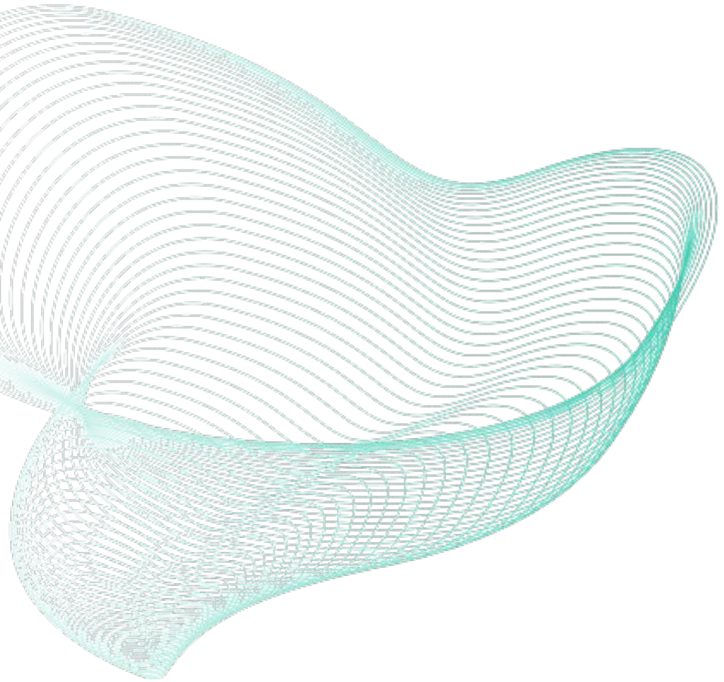
8

Conclusion



Conclusion

- Enhancing the learning experience by providing users with tools to organize, summarize, and interact with their notes in various ways.



- Implementing a tool with multiple features without compromising the accuracy as well as efficiency.

THANK YOU!