MeetingMinds: From Speech to Summary

Soorya Ram Shimgekar
sooryas2@illinois.edu
University of Illinois - Urbana Champaign

Aayush Agarwal
aayush8@illinois.edu
University of Illinois - Urbana Champaign

Suvodeep Saibal Sinha
sssinha2@illinois.edu
University of Illinois - Urbana Champaign

KEYWORDS

Text Summarization, Speech-to-Text, Query Answering, Information - Retrieval

1 INTRODUCTION

Online meetings have emerged as a key component of business communication in today's digital environment, assisting in the communication of individuals and teams that are spread geographically. The effective administration and retrieval of meeting material, however, continues to be a major difficulty. Our study implements a novel information retrieval system made especially for online meetings in order to overcome this difficulty.

Our system's main goal is to convert spoken meeting material into an organized, user-friendly manner. We do this by using a sophisticated voice-to-text model called Kaldi, which translates spoken speech into written text. This conversion is an essential first step in producing a thorough meeting transcript, which forms the basis of our subsequent processes.

Building on these transcripts, our technology generates succinct, relevant summaries in an effort to capture the essence of discussions. In order to ensure that the most important components of the meeting are precisely and effectively documented, we use Google T5 and RAKE to produce these summaries that contain decisions, action items, and key points.

Our next step is to use these transcripts and meeting notes as a special corpus for query response. Our method determines the most significant textual segments that are pertinent to the query by using cosine similarity. With the help of this feature, meeting material will be much more useful as it will become dynamic and interactive information base instead of a static record.

By transforming the way meeting content is processed, summarized, and queried, we aim to unlock new efficiencies and insights for organizations and individuals alike.

2 MOTIVATION

The rise of remote work environments in today's quickly changing workplace has significantly changed the way teams interact and communicate. The widespread use of online meetings has been essential to this change. These virtual meetings, which allow teams to interact and communicate regardless of their physical locations, have become an essential part of everyday business life. However, the transfer to this new style of communication has particular obstacles, which our initiative seeks to overcome.

One of the most significant challenges is the effective management and retention of information shared during these meetings. Important information, choices, and action items can be quickly forgotten or lost in the speed and volume of online meetings. The problem is exacerbated when team members are located in various time zones, which necessitates asynchronous communication and flexible access to meeting materials. We are driven to improve efficiency and coordination in these distant work environments so that no important data is overlooked, which is the driving force for our initiative.

The development of a private meeting content corpus creates new opportunities for knowledge management and information retrieval. Communication within a team can be greatly streamlined by providing prompt and accurate answers to questions regarding previous decisions or discussions. This feature helps to create a more cohesive and knowledgeable team environment by being especially helpful in situations where team members need to review previous discussions for background or clarification.

Our project basically aims to fill a significant gap in the tools available for remote work collaboration. Our goal is to improve online meetings' efficacy and efficiency by offering a smooth solution for recording, summarizing, and querying meeting material. Thus, improved decision-making, team alignment, and overall productivity are all supported in remote work environments.

3 METHODOLOGY

Task 1 - Speech to Text: The dataset is taken from 7.1 and an open-source toolkit for speech recognition that provides a collection of tools and libraries for building automatic speech recognition (ASR) systems is used - Kaldi[1]. Here's how Kaldi can be applied in this context:

(1) **Speech Recognition:** Kaldi can be used to transcribe the spoken content of the meeting. It can convert the audio recordings of the meeting into text by training or utilizing pre-trained ASR models. This will provide you with a textual representation of the meeting content.

(2) **Transcript Cleaning and Pre-processing:** Meeting transcripts generated by ASR systems may contain errors and inconsistencies. Kaldi can help in preprocessing the transcript to improve its quality. We will use Kaldi's capabilities to correct common ASR errors, such as misrecognition and punctuation issues. This preprocessing step is crucial for improving the accuracy of summarization.
(3) **Speaker and Text Correlation:** To enhance the summarization process, it's helpful to know who said what in the meeting. Kaldi offers tools for speaker identification, which can segment the transcript based on speakers. This information is valuable for summarization and can help attribute statements to specific participants.

Task 2 - Text Summarization: For the tasks of Text Summarization we have reviewed various papers and we choose to experiment between T5 (Text-to-Text Transfer Transformer)[2] and GPT-3.5[3] (Generative Pre-trained Transformer 3.5) as our main summarization algorithm. Based on the following requirements we decided to choose to use google's text to text transformer T5, trained on C4 dataset.

A few of the reasons to finalize T5 over GPT was that;

(1) **Cost of using:** We needed to summarize the text without having to pay and for GPT3 we had to pay to use their API, whereas Google's T5 is publicly available as opensource on Huggingface.
(2) **Time to Summarize:** Because GPT3 API required us to send the text to OpenAI's server everytime we wanted to summarize it was very slower when compared to Google's T5, to which we could locally download the T5-Base model.
(3) **Content Relevance:** Both GPT3 and T5 gave relevant abstractive summarization.

To summarize using the T5 transformer with T5-Base model we will pass a guide word in the start as following; **"Summarize: Text"**

Figure 1: T5 Text Summarization

Task 3 - RAKE algorithm: Apart from the text summarization, we thought it might be relevant to also generate important words or key phrases from the long meeting that might have been lost during the summarization phase. To achieve this, we implement the RAKE (Rapid Automatic Keyword Extraction) algorithm.

RAKE operates by analyzing the frequency of words and their co-occurrence within the text to determine their significance.

(1) **Text Preprocessing:** Tokenize the text into words or phrases. Remove stop words (common words like "and," "the," etc.) and punctuation.
(2) **Candidate Generation:**Identify potential keywords by looking at sequences of words that don't contain stop words. These sequences could be single words or phrases.
(3) **Scoring Keywords:**Calculate a score for each candidate based on its word frequency and co-occurrence. The score often considers the number of times the word appears and its appearances in different phrases. Longer phrases might have higher scores if individual words occur frequently.
(4) **Ranking Keywords:**Rank the candidates based on their scores.Higher scores indicate more important keywords or phrases.
(5) **Extracting Keywords:**Select the top-ranked keywords or phrases as the final extracted results.
(6) **Final Output:**Return the extracted keywords or key phrases, which represent the most significant terms in the original text.

Task 4 - Query Answering: The Query Answering component of our project is a critical element designed to enable efficient retrieval of specific information from the vast corpus of meeting transcripts.

(1) **Embedding Creation:** Our Query Answering methodology is based on the construction of high-dimensional vector representations of text called embeddings. We use Google's T5 (Text-to-Text Transfer Transformer) model for this. The choice of T5 is motivated by its proven effectiveness in understanding and generating natural language representations.
(2) **Data Used:** The saved text logs of the meetings serve as the main source of data for embedding creation. These logs, which are produced as a result of the speech-to-text conversion process, contain all of the meeting content and provide a comprehensive dataset for embedding generation.
(3) **Cosine Similarity:** The next step is to compare the two sets of embeddings—the transcription text and user queries—by calculating their similarity. To do this, we use the cosine similarity metric, which is a commonly used method for evaluating vector representation similarity in natural language processing.
(4) **Top Results:** The system finds the most similar entries by comparing the query embeddings with the corpus embeddings. We show the user the top 5 results that have the highest cosine similarity scores. This method

makes sure that the lengthy meeting transcripts can be searched quickly and effectively, allowing users to find the most relevant information in response to their queries.

We took practical considerations into account when deciding to use Google's T5 embeddings rather than creating our own transformer model. A large, labeled dataset is usually needed for training a custom transformer model, and creating and maintaining such a dataset can be resource-intensive. We can avoid requiring such a dataset and take advantage of a model that has been trained on a variety of text inputs by utilizing T5. In addition to streamlining our development process, this decision guarantees consistent and dependable performance when creating and comparing embeddings.

4 RESULTS

4.1 Speech To Text

We have successfully trained the speech-to-text model on dev_other split of the librispeech dataset[4]. While testing we got Error Rate as specified in Table 1. As shown in the Table 1, the model behaves good when tested on test_other split dataset of librispeech.

Dataset	WER (%)
test-clean	8.76
test-other	12.92

Table 1: Word Error Rate (WER) on different datasets.

An example of the Speech-to-Text on a real-world example is when we speak into the microphone, words, "Hello world. This is testing, one, two, three". The output from the model is shown in Figure 2, where each word is detected and printed in real time as we speak into the microphone. This demonstrates that KALDI is in fact very fast with its speech-to-text conversion and would be perfect in our use case where speech has to be converted to text rapidly. Time

Time	0.5 seconds

Table 2: Time between one word spoken and printed

to convert speech to text by KALDI is shown in Table 2.

In the actual GUI that we developed the speech-to-text is performed when the GUI shows listening. It occurs in real time as the user speaks and therefore, it's very quick. In Figure 3, the program is listening to your audio. A test audio passed as an Input speech was a 2 minute long speech about a hypothetical cancer and some hypothetical cures, etc.

After the meeting ends, the code can auto-detect the end of the meeting based on keywords like "bye", "goodbye",

```
hello
hello world
hello world this
hello world this is
hello world this is testing
hello world this is testing one
hello world this is testing one two
hello world this is testing one two three
```

Figure 2: Output from KALDI Speech-to-Text Model

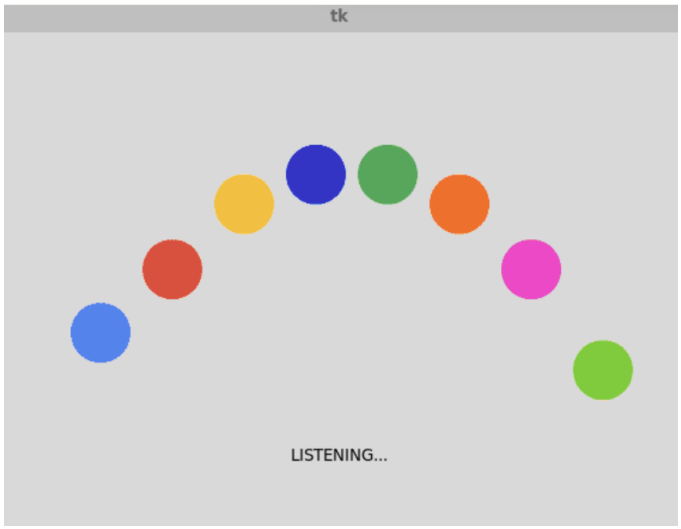

Figure 3: Listening

"farewell", etc. Once listening finishes it then goes into Processing state as shown in Figure 4

After the processing state, we see three options to choose from which are, seeing the text summary, seeing top sentences from the text, and finally, querying the text. These options could be seen in Figure 5

4.2 Text Summarization

If we press the "Text Summary" option then, based on the input from speech to text, we pass the text to Google's T5 summarizer. It limits it to a maximum of 6-7 sentences. In Figure 6

4.3 See Top Sentences

If we press the "See Top Sentences" option then, based on the input from speech to text, we pass the text to calculate the word degree scores based on the RAKE algorithm. Degree scores depend on the connectivity of a particular word and its word frequency. Based on sentences with many high-degree

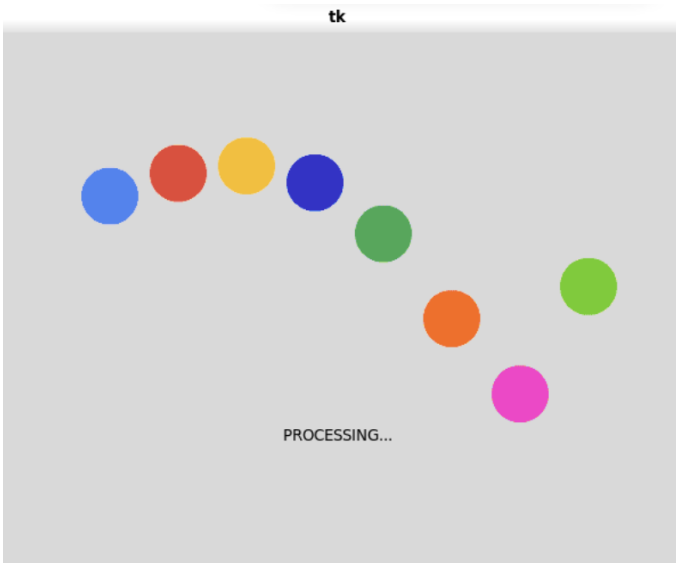

Figure 4: Processing

Figure 5: Options

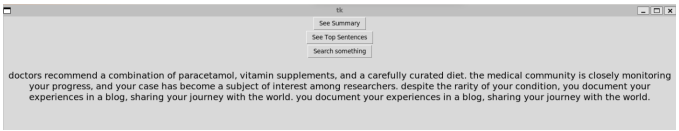

Figure 6: Summarization

score words, those words are chosen as the most important sentences. Output of this step could be seen in Figure 7

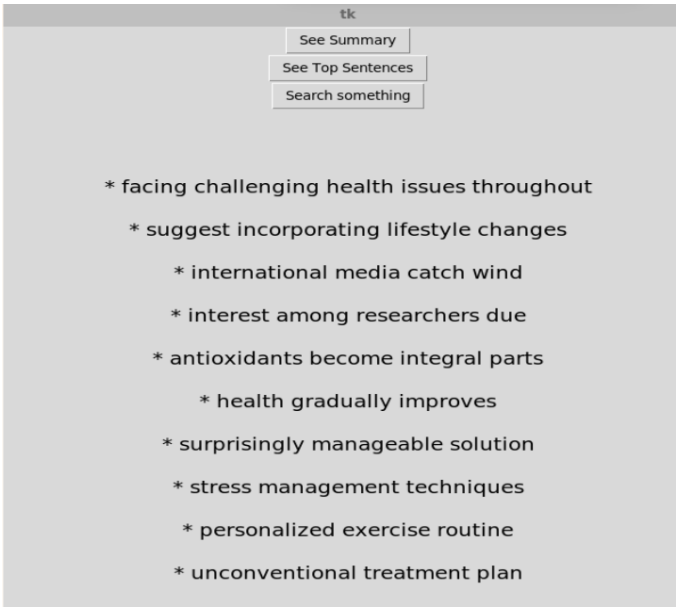

Figure 7: Top Sentences

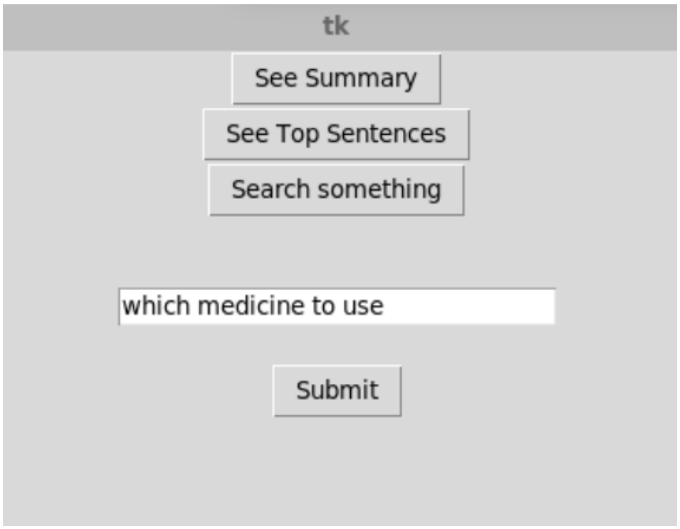

Figure 8: Query

4.4 Query and Answers

If we press the "Search Something" option then we can query anything about the meeting and we would see results relevant to the query. The query passed could be seen in Figure 8

based on the input from speech to text, we pass the text to first find the word embedding for each sentence based on the word embeddings from the T5 transformer. The length of the word embedding is 728. Then the embeddings of all the words of a sentence would be averaged to get the individual

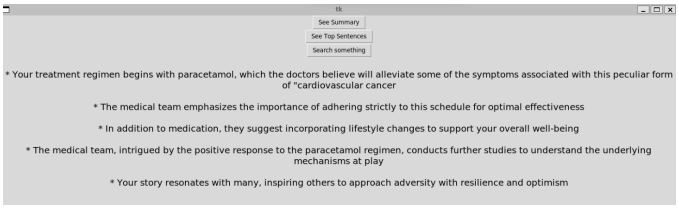

Figure 9: Result of searching for the Query

sentence embedding values. Similarly, it's calculated for the query as well. Then using cosine similarity sentences are ranked based on similarity score. The output of this step can be seen in Figure 9

5 PIPELINE

A diagrammatic representation of a brief pipeline of our project is depicted in Figure 10. Below, we have explained each of the component in detail.

(1) **Step 1: Data Collection (Audio)**
Record the speech from the microphone in real-time in a meeting
(2) **Step 2: Speech-to-Text Conversion**
Utilize the trained KALDI speech-to-text model to generate transcripts of the audio.
(3) **Step 3: Text Preprocessing**
Apply text preprocessing techniques such as tokenization, stemming, and stopword removal.
(4) **Step 4: Text Summarization**
Generate a summarized version of the transcript using Abstractive Summarization.
(5) **Step 5: Extract Action Items**
Use Named Entity Recognition (NER) or specific keyword matching to extract action items.
(6) **Step 6: Store Transcripts and Summaries**
Save these into a database as a private corpus, ensuring they are encrypted and secure.
(7) **Step 7: Display meeting summary**
Display the summary of the meeting to the user.
(8) **Step 8: Query Answering**
Use query answering method to retrieve the answer to the question asked.
(9) **Step 9: Output**
Display the answers

6 DATA

In this project, we will be using 3 key datasets

(1) ***Speech to Text (Training) - [NPTEL lecture videos [4]]*** - 1000 hours of read English speech. NPTEL lectures, encompassing a wide range of technical and academic speech.

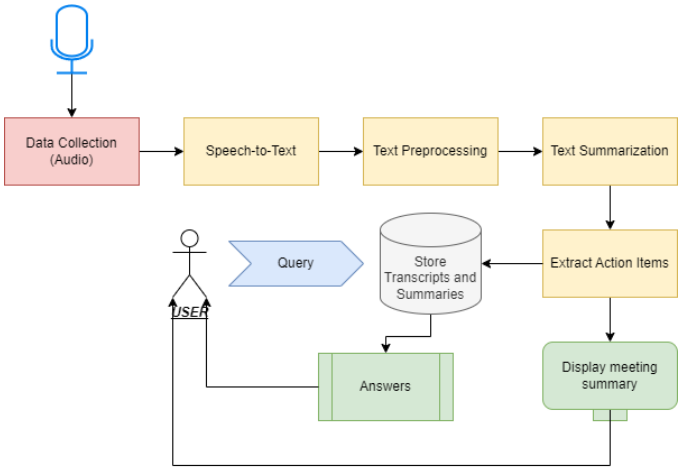

Figure 10: End-to-end pipeline

(2) ***Text Summarization - [C4 dataset [5]]*** - The C4 dataset is a comprehensive collection of diverse, web-scraped text data containing over 750 gigabytes of information, encompassing approximately 750 billion tokens and serving as a valuable resource for natural language processing research and development.
(3) ***Complete Project Testing (End-to-End) - [AMI Meeting Corpus [6]]*** - 100 hours of meeting recordings. The recordings use a range of signals synchronized to a common timeline. These include close-talking and far-field microphones, individual and room-view video cameras. The meetings were recorded in English using three different rooms with different acoustic properties, and include mostly non-native speakers.

REFERENCES

[1] Daniel Povey et al. "The Kaldi speech recognition toolkit". In: *IEEE 2011 workshop on automatic speech recognition and understanding*. CONF. IEEE Signal Processing Society. 2011.
[2] Colin Raffel et al. "Exploring the limits of transfer learning with a unified text-to-text transformer". In: *The Journal of Machine Learning Research* 21.1 (2020), pp. 5485–5551.
[3] Ashish Vaswani et al. "Attention is all you need". In: *Advances in neural information processing systems* 30 (2017).
[4] Vassil Panayotov et al. "Librispeech: an asr corpus based on public domain audio books". In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2015, pp. 5206–5210.
[5] Mahnaz Koupaee and William Yang Wang. "Wikihow: A large scale text summarization dataset". In: *arXiv preprint arXiv:1810.09305* (2018).
[6] P. von Platen, Chao Zhang, and P.C. Woodland. "Multi-Span Acoustic Modelling Using Raw Waveform Signals". In: *Interspeech 2019*. ISCA, Sept. 2019. DOI: 10.21437/interspeech.2019-2454. URL: https://doi.org/10.21437%2Finterspeech.2019-2454.