Business Meeting Summarisation System

Pallavi Lodhi, Shubhangi Kharche, Dikshita Kambri and Sumaiya Khan

May 22, 2022

Business Meeting Summarization System

Pallavi Motilal Lodhi
Electronics and Tele Comm.
SIES GST
Navi Mumbai, India
pallavilextc118@gst.sies.edu.in

Shubhangi Kharche
Electronics and Tele Comm.
SIES GST
Navi Mumbai, India
shubhangik@sies.edu.in

Dikshita Kambri
Electronics and Tele Comm.
SIES GST
Navi Mumbai, India
dikshitakextc118@gst.sies.edu.in

Sumaiya Saleem Khan
Electronics and Tele Comm.
SIES GST
Navi Mumbai, India
sumaiyakextc118@gst.sies.edu.in

***Abstract*—As the economy expands, so does the quantity of production and sales in a firm; as a result, a substantial number of business meetings are conducted day-to-day to make crucial choices. The importance of business meetings cannot be overstated. It enables you to maintain track of the organization's processes and operations to achieve the organization's goals and objectives. The business meeting findings must be kept up to date by a large number of individuals. Conventionally, people had to read long meeting reports or talk to meeting attendees to get the gist of the meeting. This summarization tool helps the user to gain the information shared in a meeting with just one click. It can be used in various domains like education, healthcare, business, etc. Existing summarization systems are limited to only the English language. This work demonstrates summarizing a business meeting held in regional or professional languages with the help of a machine learning model. The summarization is done using the abstractive method wherein words are allocated based on their frequency of occurrence in the text file. The machine learning model is connected to the NodeJS server application with the help of a python connecter. To overcome the current barriers, this system takes audio input of Hindi or English language from the user end, summarizes it using ML techniques which improve overall accuracy and provide the output in any desired language.**

***Index Terms*—Meeting summarization, Natural language processing, Abstractive summarization, Audio summarization, Artificial intelligence.**

I. Introduction

Meetings are a common tool for all types of cooperation, whether small or large. People spend hours each day in meetings, which increases the risk of missing essential information. Taking the meeting's updates is a time-consuming chore if someone misses it. The key to overcoming this difficulty is summarization. The summarization model is designed to turn long audio or text into a summarized version. The generated summary will condense the meeting's main points and assist users in catching up quickly.

The majority of existing work in the summarizing system's history is focused solely on text summarization. For the programmer, the audio summary has always been a critical task. For generating the summary of the audio, there occurs problem with the precision of the input audio. Even if the audio is synthesized, the output file will be in text format. There's a chance of losing out on some key information from the original audio because of the uncertainty with an accent. The recommended summarization system generates a meeting summary by obtaining the meeting recording from the user via a Client application developed in vanilla javaScript. The audio is recorded using speech recognition library. The input audio, which can be in any language, is summarized using a Machine Learning model that employs the seq2seq technique. The user receives both a text and an audio version of the created summary. The output can be downloaded on the client end application for future use. The summarization system process has been demonstrated in Fig. 1.

The proposed research work contains the related existing research work and background knowledge on summarization in section II. Section III contains the data collection methods. The proposed model and its comparison with other summarization models are done in section IV. The implementation procedure of the proposed system is illustrated in section V. Section VI shows the experimental input and output setup of the work. The results obtained from the proposed system are explained in section VII. The possibilities of error occurrence are analyzed in section VIII. There are a few limitations of the summarization system which are stated in section IX. Conclusion and future work is explained in section X. Section XII contains the glossary of the abbreviations used in the paper. The final section XIII contains the details of the papers referenced in this proposed work.

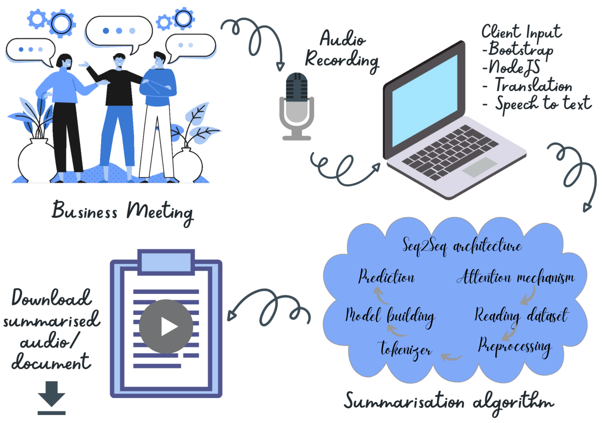


Fig. 1. summarization System

II. RELATED WORK AND BACKGROUND KNOWLEDGE

A. TEXT SUMMARIZATION

1) Abstractive summarization: The author Piotr Janaskiew et. al. [1] have proposed an abstractive method that learns an internal language representation to generate more human-like summaries in form of a comic book and provides output as text document only.

The author Deepa Anand et. al. [2] have used two major steps – First, Generation of good quality labeled data by exploiting the summary presented in the headnotes Section. Second, Utilizing such labeled data to extract the important sentences to be included in the summary. Driven semi-supervised approach to extractive legal document summarization using various neural network architectures.

2) Extractive Summarization: The author J.N.Madhuri et. al. [3] have used Automatic text summarization where the model extracts important sentences from the given data.

The author The Shi-Yan Weng et. al. [4] Presented an effective BERT-based neural summarization framework for spoken document summarization. BERT-pre trained models are very compute-intensive at interference time. If you want to use it in production at scale, it can become costly.

3) Audio Summarization: Carlos-Emiliano et. al. [5] have suggested an audio summarizing technique based on audio features, with the premise that mapping the informativeness from a pre-trained model using only audio characteristics may aid in selecting the most relevant segments for the summary.

Avneesh vartkavi et. al. [6] have used Information systems; Computing techniques; Cross-validation; Applied computing; Sound and music computing have all been employed.The model's smaller size may limit its ability to generalise to new data.

The proposed study attempts to overcome the shortcomings of previous work. The user will receive both text and audio output from this work. This work employs seq2seq modelling to address difficult linguistic issues. The extractive technique is less effective than the Abstractive method because it does not correspond to how humans summarise meetings. As a result, the suggested study uses an abstractive summarization method.

III. DATA COLLECTION

A. TEXT INPUT

As shown in Fig. 2, the user can directly provide text document of the business transcript to the system. This input is further sent to the back-end of the node application. Using ML model, the input text document is further processed. The text input can be in any regional language but in proposed work to be specific the language is Hindi. This input is translated in English before sending for further processing.

B. AUDIO INPUT

Audio input which is the prime of proposed work, this audio input has also two types either the user can record at the moment and proceed to summary or they can also use pre-recorded audio which can be any business meeting or podcast or any long audio. The input is taken using a client side interface which is a web application. In the summary section of the application, the user will have three options to provide the input to the summarize system. The user can either provide audio input to the system. The audio can be in English or can be in regional language like Hindi. If the audio is in English, it is converted to text document and provided as input to the system. If the audio is in Hindi, then first it is translated to English then it is converted to text and is provided as input. The user will have an option to start recording the meeting audio from the web application. The application uses Speech-to-Text Library to convert speech to text. This documented text is sent to the ML model for processing.

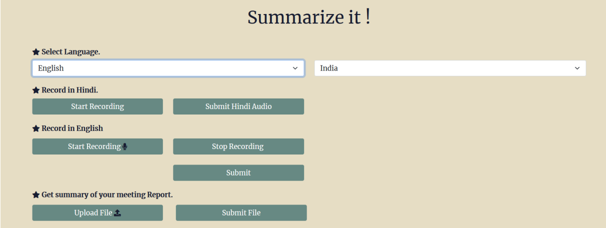


Fig. 2. Input Data from User

IV. MODEL FRAMEWORK

A. PROPOSED MODEL

There are two types of Text summarization:

1)Extractive: In this method, the trained model will extract important words/phrases from input to generate a summary.

2)Abstractive: In this method, the trained model will use its words to generate a summary. This method is far more in line with how humans summarize.

Hence system is going to use the Abstractive summarization method. In summarization,The input is a big list of words, and the output is a concise summary. So there is Many-to-Many problems at hand. So the system is going to use Seq2Seq architecture for building and training model. So, this can be modelled as a Many-to-Many Seq2Seq problem using the proposed system. seq2seq model is a special class of Recurrent neural network architectures that is typically used to solve complex language problems like Machine Translation, Text summarization.etc. There are two major components of seq2seq model: 1) Encoder 2) Decoder.

1)Encoder: An Encoder is a Long Short Term Memory model (LSTM) that reads the full input sequence, feeding one word into the encoder at each time step.

2)Decoder: The decoder is also an LSTM network which word-by-word analyses the whole target sequence and predicts the same sequence offset by one time step. Given the previous word, the decoder is trained to anticipate the next word in the sequence.

In the text summarization model, the suggested system has also used the Attention mechanism. The attention mechanism aids the model in determining how much attention each word in the input sequence requires. We can raise the relevance of certain parts of the source sequence that result in the target sequence instead of looking at all of the words in the source sequence. The attention mechanism is based on this concept.

B. COMPARISON MODEL

There are various models for text summarization:

1)Text Rank: Text Rank is a graph-based ranking model for text processing, based on Google's Page Rank algorithm, that finds the most relevant sentences in a text. Text Rank is very easy to use because it's unsupervised. The prediction will have most of the information mentioned in the original summary, as expected from an Extractive algorithm.

2)Seq2Seq Sequence-to-Sequence models are neural networks that take a sequence from a specific domain as the input and output a new sequence in another domain. Hence seq2seq is preferred over other 2 models because, the model understands the context and the key information. If you have a powerful machine, you can add more data and increase performance.

3)Bart Facebook's BART (Bidirectional Auto-Regressive Transformer) uses a standard Seq2Seq bidirectional encoder (like BERT) and a left-to-right auto-regressive decoder (like GPT). In this case, the prediction will be short but effective.

Depending on how the attended context vector is derived, there are two types of attention mechanisms:

1) Global Attention: In this mechanism, all of the source positions are given equal attention. To put it another way, all of the encoder's concealed states are taken into account while calculating the attended context vector. The system employs a global attention mechanism because all inputs are given equal weight.

2) Local Attention: The focus of local attention is on a small number of source positions. The attended context vector is derived from only a few hidden states of the encoder.

V. METHODOLOGY

A. CLIENT APPLICATION

The client application is a web application which is built on vanilla JavaScript. The system has used HTML, CSS and Javascript to build client client web pages.

1) Bootstrap: For better User Interface, proposed system is using CSS framework Bootstrap. Bootstrap is an open-source CSS framework aimed at responsive, mobile-first front-end web development.

2) EJS: Proposed system is using EJS to fetch back-end data to the client side. EJS is a basic templating language that allows you to produce HTML markup using plain JavaScript. There is no dogma about how things should be organised. Iteration and control-flow do not need to be reinvented. It's simply ordinary JavaScript. EJS is basically to run javascript on front-end by embedding it in HTML so that we don't have to write external JavaScript.

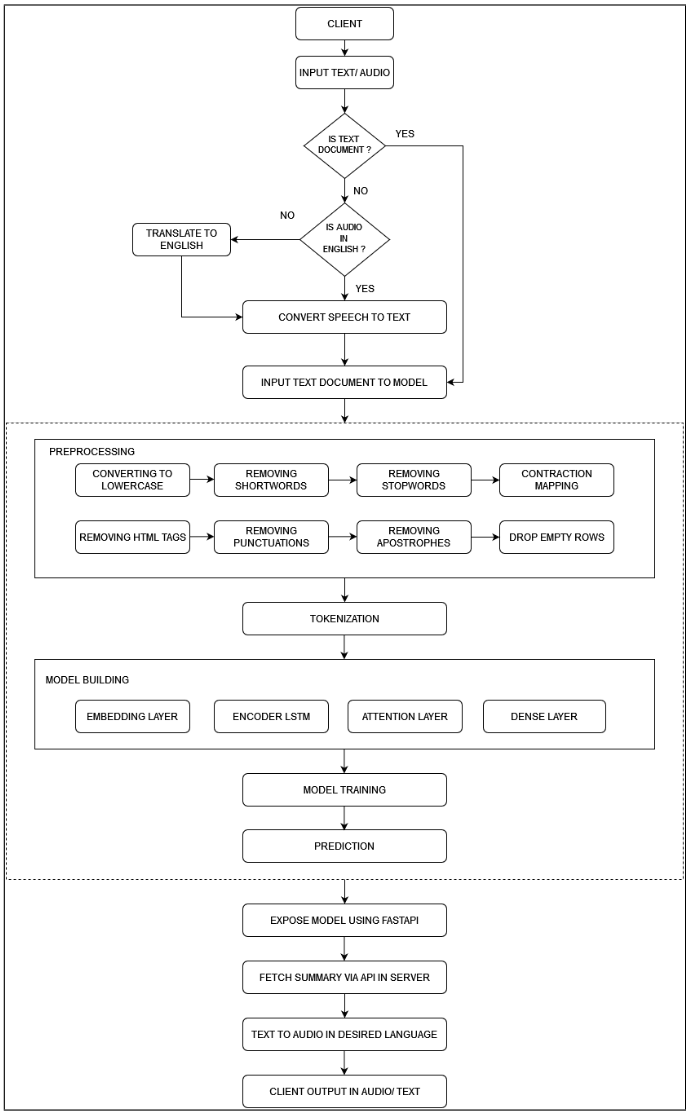


Fig. 3. System Flowchart

3) SpeechRecognition: The user has an option to record the speech directly from the browser. This speech has to be converted to text before inputting the data to the summarize system. For conversion of speech to text system is using a SpeechRecognition google library. This library is being used on the client side. It continuously listens to the audio input and save its text version into a variable. This is then saved to a text file which is provided as a input to the system.

B. SERVER APPLICATION

1) Backend Framework: The backend server is built using nodejs and expressjs. Node.js is an open source server environment. Node.js allows us to run JavaScript on the server. So as the system is using JavaScript on the client side, it is convenient to use javascript on the server side as well. Hence, we have used expressjs which is a framework of nodejs. It provides many built-in sets of functions. It is used for designing and building web applications easily and quickly.

2) Translation: After the input is received from the client side, if the input is in any regional language like Hindi, it has to be translated to English and then converted into a text document. For this translation, we are using Google Cloud Translation API. This Translation API uses Google's neural machine translation technology to instantly translate texts into more than one hundred languages. The translation is saved into a text file and which sent to the model for further processing.

C. SUMMARIZER

1) STEP 1: Imported required libraries: Pandas, sklearn, numpy,nltk these are some of the important libraries are used in System.

2) STEP 2: Reading and Cleaning data: Proposed system is using news summary dataset to train model. After reading dataset dataset is cleaned, this process is called as prepossessing. Different steps involved in prepossessing are:

- Converting everything to lowercase.
- Removing HTML tags.
- Contraction Mapping.
- Removing short words.
- Removing stop words.
- Removing Punctuation marks and special character.
- Removing any text inside parenthesis.
- Remove 's.

3) STEP 3: Creating Tokenizer: The vocabulary is built by a tokenizer, which converts a word sequence to an integer sequence.

4) STEP 4: Model building: Model is built using a 3 stacked LSTM for the encoder. By monitoring a user-specified statistic, the early stopping approach is utilised to cease training the neural network at the appropriate time. The validation loss has been tracked by the system (val-loss). When the validation loss grows, the model will stop training.

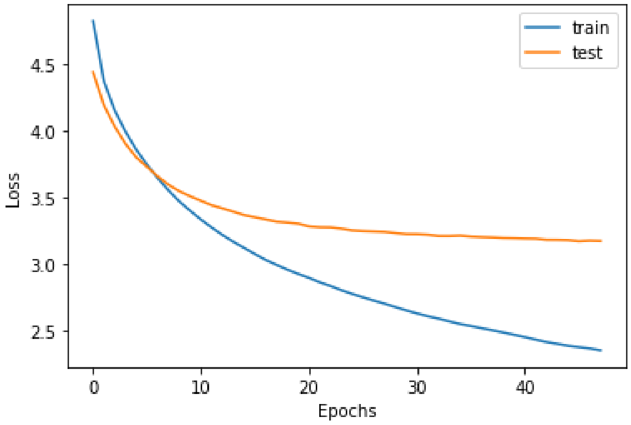


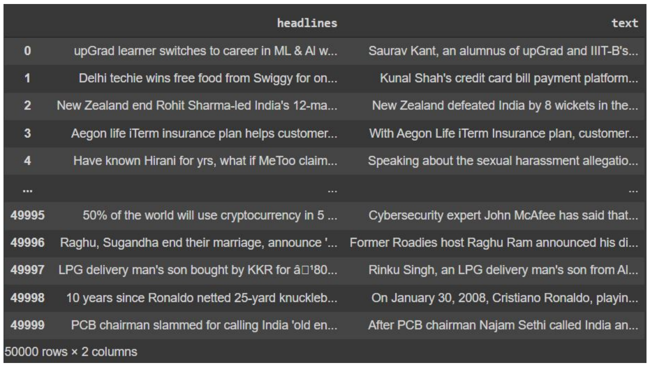
Fig. 4. Loss Vs Epochs

5) STEP 5: Prediction: System sets up the inference for the encoder and decoder to get appropriate predictions.

VI. EXPERIMENTAL SETUP

A. DATASET

In the suggested system, news summary dataset from Kaggle is used. It has 50 thousand news samples. Dataset has 3 columns. viz, headline and text(news) as shown in fig 5. In distribution of dataset we can observe that,maximum text length is between 40-80 and word count is approximately 1200. And maximum headline length is between 20-25 and word count is approximately 1200 as shown in fig. 6



	headlines	text
0	upGrad learner switches to career in ML & Al w...	Saurav Kant, an alumnus of upGrad and IIIT-B's...
1	Delhi techie wins free food from Swiggy for on...	Kunal Shah's credit card bill payment platform...
2	New Zealand end Rohit Sharma-led India's 12-ma...	New Zealand defeated India by 8 wickets in the...
3	Aegon life iTerm insurance plan helps customer...	With Aegon Life iTerm Insurance plan, customer...
4	Have known Hirani for yrs, what if MeToo claim...	Speaking about the sexual harassment allegatio...
...	...	...
49995	50% of the world will use cryptocurrency in 5 ...	Cybersecurity expert John McAfee has said that...
49996	Raghu, Sugandha end their marriage, announce '...	Former Roadies host Raghu Ram announced his di...
49997	LPG delivery man's son bought by KKR for â□¹80...	Rinku Singh, an LPG delivery man's son from Al...
49998	10 years since Ronaldo netted 25-yard knuckleb...	On January 30, 2008, Cristiano Ronaldo, playin...
49999	PCB chairman slammed for calling India 'old en...	After PCB chairman Najam Sethi called India an...

50000 rows × 2 columns

Fig. 5. Dataset Structure

Distribution of the sequence in dataset is as follows:

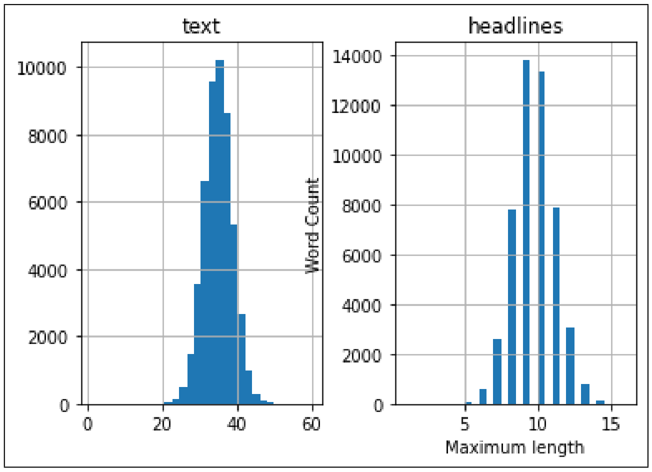


Fig. 6. Distribution of the sequence

B. INPUT

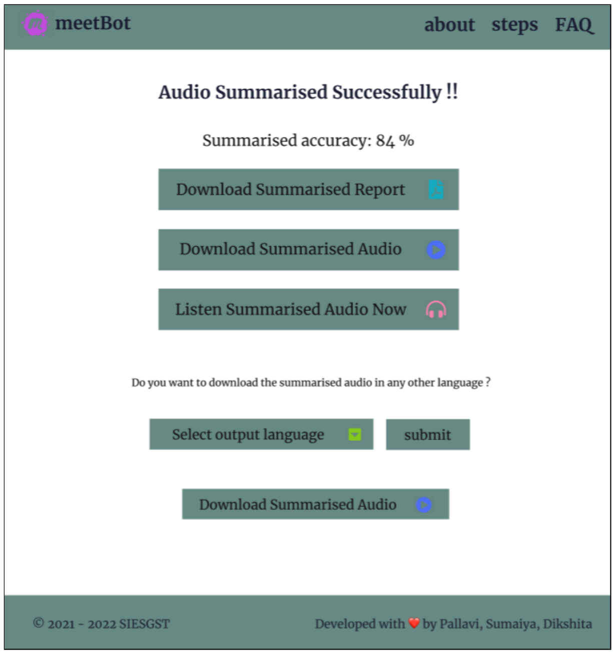
Input can be in the frontend side of the work. When user visit the website, First of all he needs to select the language, then he can record the audio by long pressing the record button. User can also upload the prerecorded file by clicking on upload file button.

C. OUTPUT

When user submit the input file, It will redirect to new web page where he will find download button to download the audio summary. There will be a button to listen audio summary on website itself.

VII. RESULTS DISCUSSIONS

The client-side application is created responsively using vanilla JavaScript. The CSS framework bootstrap is used for styling web pages.EJS and connected back-end are used on the client-side for sending the summary of the Input. The speech input is converted to a text document with good accuracy. The Hindi language audio is translated into English using Google cloud translation API which is then sent for summarization to the model. As shown in Fig. 7, the User observes a web page for downloading or listening to the summary after submitting Input.



Audio Summarised Successfully !!

Summarised accuracy: 84 %

Download Summarised Report
Download Summarised Audio
Listen Summarised Audio Now

Do you want to download the summarised audio in any other language ?

Select output language
submit
Download Summarised Audio

© 2021 - 2022 SIESGST

Developed with ❤ by Pallavi, Sumaiya, Dikshita

Fig. 7. User Output

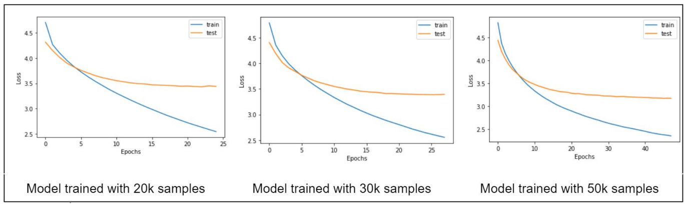
The input text file is summarized by following the steps like pre-processing, tokenization, model training and building. Abstractive summarization is used in the model. Because it does not simply copy key phrases from the source text but also potentially come up with new phrases that are relevant, which can be seen as paraphrasing. Hence, output summary is nearly similar to the exact interpretation of the meeting discussion.

Initially, the model is trained with 20k samples and parameters are observed as shown in Table 1. To obtain meaningful summary, the model is trained again with 30K and 50K samples and better results are observed as shown in Table 1.

Table 1. Model data

No. of samples	20K	30K	50K
Percentage of Rare Words	60	58	57
Size of Vocabulary	14587	17834	22463
Training Samples	18894	27893	44993
Validating Samples	2100	3100	5000

As shown in Fig.8, Early stopping occurs at the 25th epoch,28th epoch, and 48th epoch for 20K, 30K, and 50K samples, respectively. Hence, with increasing number of samples, the model overfitting probability also reduces as the model gets exposed to more vocabulary leading to meaningful output summary, as shown in Fig.9.



Model trained with 20k samples

Model trained with 30k samples

Model trained with 50k samples

Fig. 8. Model comparison

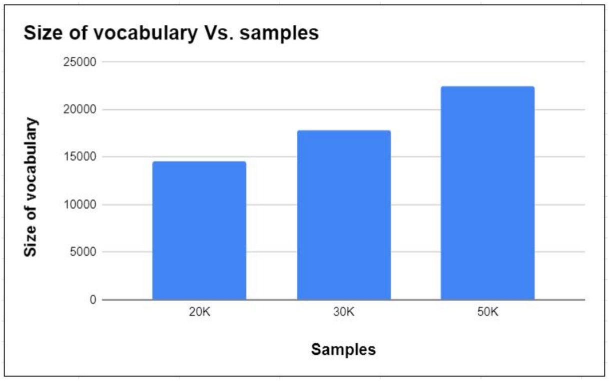
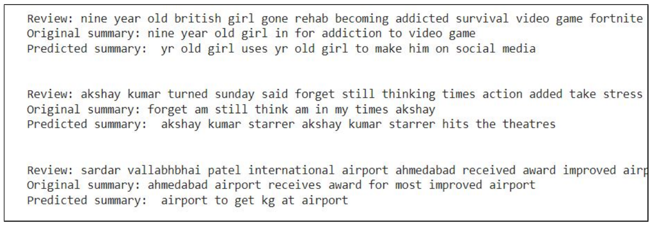


Fig. 9. Vocabulary size vs Samples

The user can get the text summary in whichever language he or she wants. This output is available in two formats: audio and text. Fig. 10 depicts the Trained summarization model's expected summary.



```
Review: nine year old british girl gone rehab becoming addicted survival video game fortnite
Original summary: nine year old girl in for addiction to video game
Predicted summary:  yr old girl uses yr old girl to make him on social media


Review: akshay kumar turned sunday said forget still thinking times action added take stress
Original summary: forget am still think am in my times akshay
Predicted summary:  akshay kumar starrer akshay kumar starrer hits the theatres


Review: sardar vallabhbhai patel international airport ahmedabad received award improved airp
Original summary: ahmedabad airport receives award for most improved airport
Predicted summary:  airport to get kg at airport
```

Fig. 10. Predicted Summary

VIII. ERROR ANALYSIS

There are few reasons which might affect the summary output obtained by providing the input audio file.

A. APIs CONNECTION

The suggested work uses GTTs library for recognising the speech audio and translating it to text and cloud translation API to translate Hindi audio to English. These are Cloud libraries, if the internet connection is not stable, the GTTs library may not be able to read and record the audio.

B. AUDIO INPUT DISCREPANCY

If the words aren't spoken correctly, the translation from audio to text will not be done efficiently. For Hindi to English audio translation, if the Hindi words are not known by the cloud translation library, it might not translate the audio properly.

C. OTHER LANGUAGES USED

If the languages which are used in the business meetings are any other than English or Hindi, the summarization system will not be able to translate the language and convert it to text document to input the file to the system. So the languages used in the meeting should be strictly English or Hindi.

IX. LIMITATIONS

A. BROWSER LIMITATIONS

The current system architecture includes Google cloud libraries which are GTTS for Speech to text conversion and Google Cloud Translation API for translating Hindi audio to English. These libraries for recognising the speech works only on Chrome browser, it does not work on any other browsers like firefox or edge, etc.

B. LANGUAGE LIMITATIONS

The proposed work is to summarize regional language as well. Currently the system is only summarising in Hindi and English language. If any other languages like Marathi, Gujarati are used in the meeting, the system will not recognise the language and will cause discrepancy in the summarization.

X. CONCLUSION AND FUTURE WORK

The purpose of this work is to provide highly accurate audio summary by analyzing the content of the input file. It has been proved that audio summarization can produce a more structured and diverse summary, spanning from trends to topics. The APIs and libraries used throughout collectively help in obtaining the high end output

As there are daily new research and development in technologies. So, in future we can expect betterment in above work, such as by allowing the user to choose the length of text summary and duration of audio summary. With that, an improved training model with a larger number of samples than the previous study for developing huge vocabularies which will effectively improve the performance of the system. The proposed work only uses Hindi regional language; however, many more regional languages can be included in further update.

XI. GLOSSARY

Acronym	Reference Abbreviation
JS	JavaScript
ML	Machine Learning
EJS	Embedded JavaScript
NLP	Natural Language Processing
CSS	Cascaded Style Sheet
UI	User Interface
RNN	Recurrent Neural Network
GRU	Gated Recurrent Neural Network
LSTM	Long Short Term Memory
GUI	Graphical User Interface
API	Application programming interface
Seq2Seq	Sequence to Sequence

REFERENCES

[1] Piotr Janaskiew icz, Justyna ktysinska, Marcin Prys, Text Summarization For Storytelling,January 2, Available:https://www.researchgate.net/publication/327292982

[2] Deepa Anand a, Rupali Wagh, Effective deep learning approaches for summarization of legal texts ,2019, Available:https://www.sciencedirect.com/science/article/pii/S1319157819301259

[3] J.N.Madhuri, Ganesh Kumar.R, Extractive Text Summarization Using Sentence Ranking,2019, Available:https://ieeexplore.ieee.org/document/8817040?denied=

[4] Shi-Yan Weng, Tien-Hong Lo, Berlin Chen , An Effective Contextual Language Modeling Framework for Speech Summarization with Augmented Features ,2021, Available:https://arxiv.org/abs/2006.01189

[5] Carlos-Emiliano González-Gallardo, Audio Summarization with Audio Features and Probability Distribution Divergence, 20 Jan 2020, Available: https://arxiv.org/abs/2001.07098

[6] ANEESH VARTAKAVI and AMANMEET GARG, PodSumm:Podcast Audio Summarization, 2020, Available:https://www.researchgate.net/publication/344347201

[7] Ming Zhong, Da Yin, Tao Yu, A New Benchmark for Query-based Multi-domain Meeting Summarization, 2021, Available:https://arxiv.org/abs/2104.05938

[8] Abdullah Aman Khan, Jie Shao, Content-Aware Summarization of Broadcast Sports Videos: An Audio–Visual Feature Extraction Approach, 04 February 2020, Available: https://link.springer.com/article/10.1007/s11063-020-10200-3

[9] Aneesh Vartakavi, PodSumm – Podcast Audio Summarization, 22 Sep 2020, Available:https://arxiv.org/abs/2009.10315

[10] T.-E. Liu et al., A hierarchical neural summarization framework for spoken documents, 2019 in Proc. ICCASP.

[11] Devlin J. Chang et al., Bert: Pre-training of deep bidirectional transformers for language understanding., in Proc.NAACL-HLT, pp. 41714186, 2021

[12] C. Qu, et al., BERT with history answer embedding for conversational question answering, arXiv preprint arXiv:1905.05412, May 2019, Available: https://arxiv.org/abs/1905.05412

[13] L. Lebanoff et al., Scoring sentence singletons and pairs for abstractive summarization, arXiv preprint arXiv:1906.00077, May 2019, Available: https://arxiv.org/abs/1906.00077

[14] Y. Gu and Y. Hu, Extractive summarization with very deep pretrained language model, International Journal of Artificial Intelligence and Applications, vol. 10, no. 2, pp. 2732, March 2019, Available: https://www.researchgate.net/publication/332293134

[15] H. Zheng and M. Lapata, Sentence centrality revisited for unsupervised summarization, arXiv preprint arXiv:1906.03508, 2019, Available: https://aclanthology.org/P19-1628/

[16] J.-T. Chien, Hierarchical Pitman-Yor-Dirchlet language model, IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 23, no. 8, pp. 12591272, 2020, Available: https://ieeexplore.ieee.org/document/7098357