International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Jul-2022)

https://dx.doi.org/10.12785/ijcds/120110

Automated Generation of Meeting Minutes Using Deep Learning Techniques

Megha Manuel[1], Amritha S Menon[1], Anna Kallivayalil[1], Suzana Isaac[1] and Lakshmi K.S[2]

[1]*Department of Information Technology, Rajagiri School of Engineering and Technology, Kochi, India*
[2]*Assistant Professor,Department of Information Technology, Rajagiri School of Engineering and Technology, Kochi, India*

Received 2 Jun. 2021, Revised 26 Mar. 2022, Accepted 15 Jun. 2022, Published 1 Jul. 2022

Abstract: Meeting minutes are important to keep track of key decisions and agreements that were made during a meeting. It is crucial to document the topics discussed and the decisions made so they can be reviewed at the beginning of the next meeting as well as for future reference. Many companies while conducting meetings, keep paid employees to note down meeting minutes taking up valued time and resources. We offer a solution to make better use of available tools and technological advancements to help employees conduct effective discussions to boost a company's productivity. Automated Minute Book Creation (AMBOC) uses machine learning to derive key information from important discussions. AMBOC is an automated system to create transcripts and minutes of a meeting with the added advantage of speaker recognition. The model we propose will be capable of transforming an audio file into plain-text using Deep Neural Networks (DNN), recognizing the speaker using Mel Frequency Cepstral Co-efficient (MFCC) as well as summarizing the meeting transcript into condensed minutes with the help of Transformers.

Keywords: Speech Recognition, Speaker Verification, Text Summarization, Mel Frequency Cepstral Coefficient (MFCC), Dynamic Time Warping (DTW), Transformer

1. Introduction

Meetings have played a crucial role in many organizations, for a multitude of reasons such as brainstorming, planning, problem solving and decision making. A meeting can be defined as two or more people coming together to effectively communicate and discuss ideas and make decisions. The earliest evidence of people coming together for a meeting date back millennia, all the way back to 29BC when Julius Caesar created Julius Curia, where Curia in modern English translates to a gathering of men or meetings.

According to a survey by Booqed in 2019; on average an employee has eight to twelve meetings a week, each meeting having a duration of 30 minutes to an hour. Meeting minutes are a vital part of having efficient meetings. They serve as a reference for a later point of time as well as a reference for those absent from the meeting. They can also be used as corporate defense in particular situations where documented proof is required.

Over the last few decades, technology has taken the world by storm. Despite how long and how often many of us are attending meetings there hasn't been any prominent advancements in trying to make minute booking efficient. In most cases, organizations either externally hire a scribe or ask one of the participants to jot minutes during a meeting. This is a blatant waste of company resources and finances. In this study we propose a model that will be able to automatically create meeting minutes using only the meeting recording as an input. Our proposed model for AMBOC is partitioned into three smaller models;speech-to-text, speaker verification and text summarization. AMBOC was primarily designed to work with English recordings.

Although, as we are using Google API for speech to text and text summarization, AMBOC supports up to 119 languages. Languages such as Hindi, Arabic and Bosnian. Speech-to-Text also known as speech recognition will transcribe an audio file such as a wav file into a text file of what was spoken in the recording. Speech-to-Text models analyze the audio, detect phonemes and use syntax and context to generate a transcript of what was being said. Speech recognition is already a highly developed and vastly used technology.

Speaker verification on the other hand is used to identify who the speaker is. Speaker verification can be classified as text independent and text dependent models [2]. In text dependents models the speaker must identify themselves by saying a phrase that uniquely identifies them such as an employee ID or their name. Whereas text independent

E-mail address: meghamanuel26@gmail.com,amritha3ammu@gmail.com, ajkallivayalil@gmail.com, suzanaisaac@gmail.com, lakshmiks@rajagiritech.edu.in

models require no assistance from the speaker as they are not required to say any unique phrase. As text dependent models are more demanding and inconvenient for the user, we will only be discussing text independent models in this paper as it is more suitable for this study.[3]

The final model, Text summarization converts a lengthy amount of text into a compressed summary. Text summarization can be broadly classified into two categories; extractive and abstractive. In extractive text summarization, key words and phrases are found and a subset of the most important lines are given as the summary. While abstractive summarization consists of generating new sentences that highlight the key themes of the text. Throughout this paper we will only be discussing abstractive text summarization models as it is more appropriate for our system.[4]

For these three models; speech recognition, speaker recognition and text summarization-there are a variety of machine learning algorithms that can be chosen to build our proposed system. In this paper we will initially discuss similar studies such as the one done by Rachman and Khodra to create minutes for a meeting spoken in Indonesian [5] . We will then move on to compare a range of algorithms for each of the three models. The machine learning algorithms we will be discussing are; Hidden Markov Model(HMM) and Deep Neural Networks(DNN) for speech to text, transformer and LSTM-CNN for speaker recognition and MFCC with Dynamic Time Warping and Transfer Learning for text summarization. We will firstly discuss the different algorithms and then move on to comparing and analyzing them to discover which algorithms are best suited for our proposed system, Automated Minute Book Creation (AMBOC).

2. LITERATURE REVIEW

The meeting recording is harnessed to identify the key points of the meeting. There are various papers dealing with automation of hoziyah Haitan Rachman and Masayu Leylia Khodra have worked on a similar field as they were trying to find meeting minutes in Indonesian language. They used rhetorical sentence categorization for retrieving significant data of the meeting minutes. They experimented using SMOTE and resampled the existing terms to balance the instances per class. Naive Bayes, SVM Linear, IBk, and J48 tree are the four classifiers used in their experiment. Their model was trained with a transcript dataset. Their model uses SMOTE as well as 10-fold cross- validation in IBk classifier. This attained an F- measure of 85.22 percentage and resampling the model attained 94.52 percentage. The meeting minutes were not considered from the result of speech. The final output was attained without speaker recognition [5].

Another paper discusses an experiment conducted for finding meeting minutes in parliamentary speeches. This was done by Justin Jian Zhang, Pascale Fung, and Ricky Ho Yin Chan. Their paper uses a single classifier called Conditional Random Field (CRF) to convert text into chunks and extract salient features of the text. A rhetorical syntax tree helped each chunk to be represented in a tree structure which makes it easier to use classifiers for extracting prominent features in a sentence. The accuracy achieved in this experiment was 73.2 percentage when using the ROGUE-L F-Measure. ROGUE-L F-Measure calculates the match distance measured in degrees between the extracted summaries of the speech and the minutes that was referred. The parliament speeches are planned which implies that it cannot give accurate results for unplanned meetings like regular office meetings. The meeting minutes used extractive summarization which has poor syntactic analysis and world knowledge. Moreover, CRF is computationally complex during the training of the algorithm which makes new data entry availability difficult to re-train the model [6]. Beam Tasbiraha Athaya, Sirajum Munira, Afsana Zaman, Syed Akhter Hossain and Col. A B M Humayun Kabir are authors of a research they conducted to automate meeting scheduling and managing documentation of the meetings. They used Base64 algorithm to encode the content and size of the files stored in their database. A secured key is used by the employees to decode the content and only members in that organization will be able to use the document. The limitation of their work is that it cannot manage image and video files [7].These studies are shown in Table I.

3. PROPOSED METHODOLOGY

A. System Architecture

The AMBOC architecture diagram shown in Figure 1. consists of three different modules. This includes speaker verification, speech-to-text conversion and text summarization before obtaining the final output in the form of summarized meeting minutes. Sample recording of speakers along with recordings of meeting are passed as input to speaker verification module while recordings of meetings are being passed as input to speech-to-text conversion module. Both these inputs are passed as wav files. In speaker verification module, the speakers are identified based on their voices. Feature vectors of each voice recording is created using MFCC (Mel Frequency Cepstral Coefficients) and these are then tested for matched pair and verified using DTW (Dynamic Time Warping) algorithm where two feature vectors of sample recording and meeting recording of speakers are compared to verify identity of speaker. In speech-to-text module, the meeting recordings get split into chunks based on silence and are then gets converted to text. The combined output from speaker verification and speech-to-text module, form the meeting transcripts. The output from speech-to-text module then gets passed to the text summarization module where we use T5 transformers that applies abstractive text summarization to obtain summarized meeting minutes eliminating the unrequired stop words and content.

B. Speaker Verification

Speaker verification is the authentication of the identity of a person from characteristics of their voice. The speaker verification system is divided into two phases, enrollment

TABLE I. Algorithms or Classifiers used by Researchers used in Meetings

Sl No	Researchers	Algorithm or Classifiers used in their work	Year
1	Tasbiraha Athaya, Sirajum Munira, Afsana Zaman, Syed Akhter Hossain, Col. ABM Humayun Kabir [5]	Base64 algorithm to encode the content and size of the files	2018
2	Ghoziyah Haitan Rachman, Masayu Leylia Khodra [6]	Rhetorical sentence categorization for retrieving significant data in meeting minutes. Experimented using the SMOTE. Naive Bayes, SVM Linear, IBk, and J48 tree are the four classifiers used in their experiment.	2016
3	Justin Jian Zhang, Pascale Fung, Ricky Ho Yin Chan [7]	Conditional Random Field (CRF) classifier is used to extract salient features of the parliamentary speech and ROGUE-L F- Measure is to examine the match between extracted summary and minutes to be referred.	2011

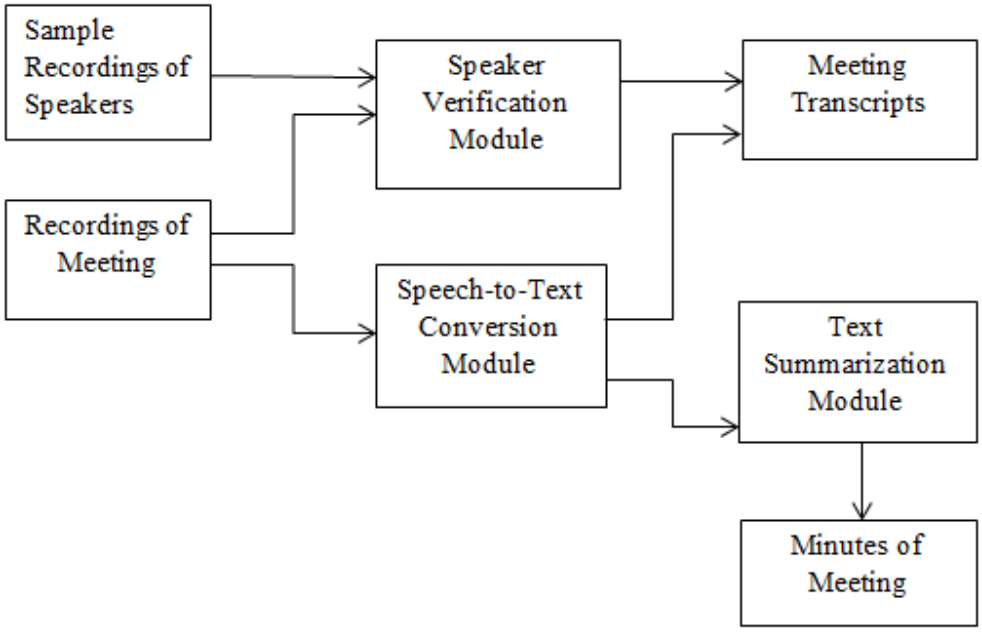

Figure 1. Architecture Diagram of AMBOC

and verification. An enrollment profile is made using a small audio recording of the speaker and a unique voice signature is made for each speaker. It is then verified by comparing input voice samples to a list of enrolled voices. Speaker verification can be text- dependent or text-independent. We use text-independent verification as speakers have no restriction on what they want to speak as voice features are extracted to check if they belong to same speaker. This system is often used in speaker verification as it relies on the vocal tract of the speaker. It helps in better protection against imposters and does not limit the content the speaker can speak. [8]

1) Feature Extraction

The core task of all speaker recognition systems is to extract vectors that represent speaker voice characteristics which can then be used to recognize specific speakers. The Mel Frequency Cepstral Coefficients (MFCC) is widely used in Speaker verification systems for extraction of frequency information from raw wav files so that the speech recognizer will be less influenced by the presence of noise or reverberation [9]. MFCC is based on knowing the frequency variance of the human ear. Feature extraction involves obtaining different features such as power, pitch, and vocal tract configuration from the speech signal Steps involved in MFCC are Pre-emphasis, Windowing, FFT (Fast Fourier Transform), Mel filter bank and DCT (Discrete Cosine Transform) computation. For extracting features with MFCC, The first step is to apply a pre-emphasis filter on the signal to amplify the high frequencies. Next step is windowing which involves slicing of the audio signal into sliding frames. It has attributes Winlen (length of analysis window in seconds, having a default value of 0.025 seconds) and Winstep (step between successive windows in seconds, having a default value of 0.01seconds).Then Fast Fourier Transform (FFT) is applied to find power spectrum of each frame. Filter bank processing is carried out on the power spectrum, using mel-scale. DCT (Discrete Cosine Transfer) is applied to the speech signal after converting power spectrum to log domain to calculate MFCC coefficients.[10] Lower frequencies are given more resolution compared to the higher frequencies similar to how human ear functions and this is mapped using the Mel filter bank formula given as,

$$\text{Mel(f)} = 1125 * \ln(1+f/700)$$

or

$$\text{Mel (f)} = 2595 * \log (1+f/700) \quad (1)$$

As Here, Mel(f) denotes the frequencies on the Mel-scale measured in Mels and f on the R.H.S denotes normal frequencies measured in Hz. We considered two approaches for our speaker verification system. These are Transfer learning and DTW algorithm.[11]

2) Transfer Learning

In this approach, speakers are verified using less amount of training, so by using only a few words or sentences, we can achieve high accuracy rate. All audio files in wav format are split into 5 second segments and then converted to spectrogram. CNN is trained on speakers as feature extractor, and the last fully connected layer is cut off and its output is fed into an SVM. This approach is known as transfer learning. It combines the features of both CNN and SVM to provide reasonable performance. The main layers of CNNs are convolutional layer, a pooling layer, and a fully connected layer. The convolutional layer is most important part of CNN model, this layer is used to extract features from input and to pass it to next layer.[12] The convolution layer is calculated as follows :

$$x_j^l = f(\Sigma_{i \in M_j} x_i^{l-1} * k_{ij}^l + b_j^l) \quad (2)$$

Where x_i^{l-1} is the characteristic map of the output of the previous layer, x_j^l is the output of the ith channel of the jth convolution layer, f (.) is called the activation function. Here M_j is a subset of the input feature maps, k_{ij}^l is a convolution kernel, and b_j^l is the corresponding offset. The limitation of this model is that the accuracy and performance declines when tested on the same gender groups or as size of group increases. For AMBOC, we used the DTW algorithm as it produces better results. It also doesn't get error while identifying speakers of same gender and is more accurate.

3) Dynamic Time Wrapping (DTW) Algorithm

All humans have diverse sound characteristics and the best method to identify each speaker from characteristics of their voice is by using Dynamic Time Warping (DTW) algorithm. DTW compares the given voice samples. There will be a sample and test audio and the distance between feature vectors of these audio are compared to identify to which speaker the voice belongs to. The minimum distance and minimum index values are used to identify the identity of the speaker. DTW is widely used method for speaker recognition.Implementation of DTW is shown in Figure 2.

All humans have diverse sound characteristics and the best method to identify each speaker from characteristics of their voice is by using Dynamic Time Warping (DTW) algorithm. DTW compares the given voice samples, there will be a sample and test audio and the distance between feature vectors of these audio are compared to identify to which speaker the voice belongs to. The minimum distance and minimum index values are used to identify the identity of speaker. DTW is widely used method for speaker recognition.[13] DTW calculates the optimal warping path between two feature vectors of audio. Warping path is distance between two feature vectors, the smaller the warping path, more chances to identify the owner of voice. Also, two words can be spoken by same person in different speeds. For example, hello can be pronounced as 'hello' or 'heeeeelloooo'. DTW solves this problem by arranging these words correctly and finding the minimum distance

between two words, to check if they belong to the same speaker or not [14].

The feature vectors of the speaker's sample and test audio are obtained using MFCC, which has a high accuracy rate. And then both these features are compared based on distance using DTW algorithm and the owner of the voice can be identified. For two feature vectors x and y, d(x,y) is the distance between them e.g. d(x,y) = mod(x-y). The formula for computing DTW is shown below:

DTW [i, j]=cost + minimum (DTW [i-1, j], DTW [i, j-1],DTW [i-1, j-1]) (3)

The complexity of computing DTW is O (m * n) where m and n represent the length. The Python Speaker verification toolkit is used. Feature vectors of sample and test audio (in wav format) are created using MFCC. It is then tested and the distance between them is calculated using the DTW algorithm. The index of the element from the list of voices that represents the nearest voice data is returned as output. [15]

C. Speech-to-Text

After speaker verification, the next process is voice to text conversion. One of the more effective approach is the Hidden Markov Model (HMM).It has fewer assumptions of independence also it does not assume that the probability that sentence i is in the summary is independent of whether sentence i-1 is in the summary. Furthermore, we use a joint distribution for the features set, unlike the independence-of- features assumption used by naive Bayesian method. For the HMM approach, the sentences are chosen with the maximum posterior probability of being a summary sentence. For a summary of length k, we choose the sentences with the k largest values of gt [16].

$$W == argmax(P(W)P(Y/W)/P(Y)) = argmax(P(W)P(Y/W)) \quad (4)$$

The HMM's have low processing abilities of computer chips and they have problems when incorporating the huge size of training data. Hence, we use the DNN method since it has better accuracy compared to classical methods [17]. In this algorithm, we used the Speech Recognition module of Google, which relies on a Deep Neural Network model. The ease-of-use and flexibility of the Speech Recognition package makes it an excellent choice. This provides a model running in minutes hence can avoid the process of building a model from scratch by training the dataset. Google Web Speech API— provisions a default API key that is encoded into the Speech Recognition library, and that is the particular recognizer function used in AMBOC. The primary purpose of a Recognizer instance is to recognize speech; hence each instance comes with a variety of settings and functionality for recognizing speech from an audio source. [18] As the size of the meeting audio file increases, the accuracy of Google speech recognition API for speech recognition decreases. Therefore, there is a need to process the .wav file into smaller chunks meaning smaller units of audio files in .wav formats and then feed these chunks one by one to the recognizer module [19]. It is done by splitting files into chunks of constant size. For the AMBOC module, we have split the audio file into chunks of size 10 sec. After processing these files in the recognizer module, the results of all these chunks have been concatenating. A problem that arises during performing the above method is when splitting the audio file into chunks of constant size might interrupt sentences in between and might lead to loss of important words because google will not be able to recognize broken or incomplete words.

To avoid this problem in the AMBOC module, we have used the method of splitting the audio file based on silence. It is based on the automatic pause that is created by humans during speech. By splitting the audio file into chunks based on these silences, we can process the file sentence by sentence and concatenate them to get the result. This method is more precise than the previous method because we do not cut sentences in between and the audio chunk will contain the entire sentence without any interruptions. This way, we do not need to split it into chunks of constant length.[20]

Splitting is done by passing the files into the function splitonsilence(),which has attributes of the audio file stored in .wav format , minsilencelen which specify the minimum amount of silence needed between the sentences to get splitted into sentences and silence thresh mean the upper bound for how quiet is silent in dFB . Each chunk formed by this method is further passed into the function adjust-forambientnoise() for removing the noise. And later into recognizegoogle() for reading the speech. The speech read would be written down into variables.[21].

D. Text Summarization

Summary construction is a complex task which ideally would involve deep learning techniques for processing. An attractive summary is a subset of content, where it would convey the overall meaning. Greatest advantage of this tool is it gives a summary which can reduce the time in reading reports. Before the content gets summarized it is important to pre- process the content that is derived from voice to text module. The main steps of pre- processing are: i) stop word elimination: common words which are not relevant to convey the meaning of text such as the, are, etc are removed. ii) lower case: entire text would be converted to lower text. iii) stemming: finds words with syntactically similar meaning words, such as plural and brought to its radix form.[22] The quality of a generated summary is the key point in evaluation of summarization. TF-IDF model is one of the most popular techniques for extractive text summarization. The idea behind this method is that, when a word appears frequently in a document, then such words are considered as important and are given a high score [23]. If the same word appears in too many other documents, it's

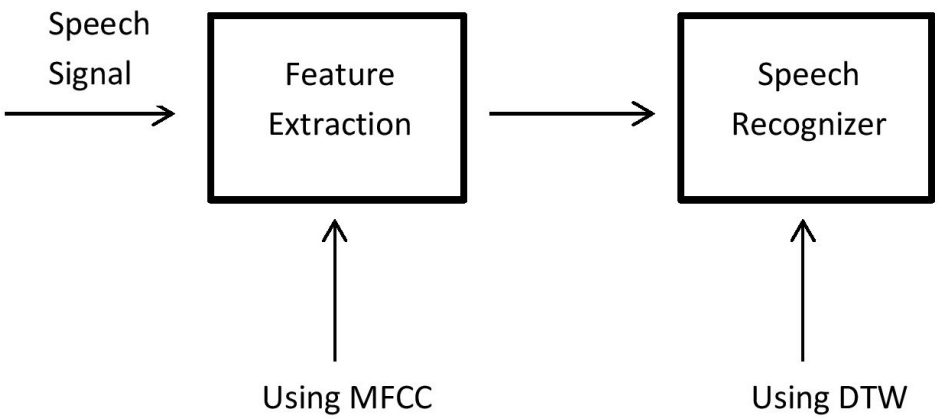

Figure 2. Shows the implementation of DTW algorithm.

probably not a unique identifier, therefore we should assign a lower score to that word. Mathematical concept:

TF(w) = (No of words w appear in document / Total no of words in the document)

$$IDF(w)=\log_e(Totalnoofdocument/Noofdocumentwithw)$$

TF IDF(w) =TF * IDF (5)

Since AMBOC is mainly used in creation of a summary for meeting recording, the importance of the words analysed using this process does not show a high rate of accuracy since there may be use of technical terms that are specific to certain departments. Those words' importance would be hard to evaluate, using the current set of datasets.

Another approach for text summarization is abstractive summarization shown in Figure 3 [23]. The algorithm used in this approach is called T5 Transformer. It is a model trained with the help of google in an end-to-end manner. This uses LSTM structure to gain the required output. Here the conceptual meaning is read by the encoder and decoder creates new words by taking the overall summary using the help of google pathways. To use this model, there are parameters like number of characters in text, the size of required summary, and number of times you wish to repeat the same word. The summary obtained through this method is more specific to the required manner the end user desires [24][25]. Another features of t5 transformers include converting from one language to another language, question answering, regression, binary classification and multi-label classification . T5 transformers works for most of the famous languages in case of summarization. By using features like regression and binary classifications, analogy or the hidden contents could be found. But as the complexity of text given for processing and its requirements increases it would negatively impact the accuracy of the entire model.[26]

For summarizing both the model and the tokenizer would be generated beforehand using t5-small tranformers. Hence model with be pretrained with words of required languages. Each chunk that have been converted into text will be concated with a word "Summarize:" in the beginning of the text and passed into the tokenizer.encode(), as the attributes. [27] The output generated will be in the form of a vector with tokens. To generate the summary, these tokens are passed into the function generate(), using the pretrained model.Then the output generated from encoder is passed into the decoder to convert it into readable format.[28]

4. ALGORITHM

1) Start
2) Samples and meeting recordings are pre-processed by converting all the files into .wav format and saved. All the sample voices are inserted into an array called Samples [ ] of size m(no: of sample voices).
3) The recording is split using speaker diarization. The split chunks are stored as .wav format in an array named Recording[i], where i ranges from 0 to n.
4) Initialise i to 0.
5) Call function voicerecognition() for parameter Recording[i], and then store result in k[i].
6) Increment value of I, go to step 5 until I equals n.
7) Initialise i to 0.
8) Call function voicetotext() for Recording[i]
9) Write Samples[k[i]] into a text file 'meeting'.
10) Write the output of voicetotext() into 'meeting.txt' and to 'temporary.txt'.
11) Pass the file temporary.txt as the parameter for the function text summarization().
12) The output of the above function is stored into 'AMBOC.txt'
13) Increment the value of i, and go back to step 8 until i equals n.

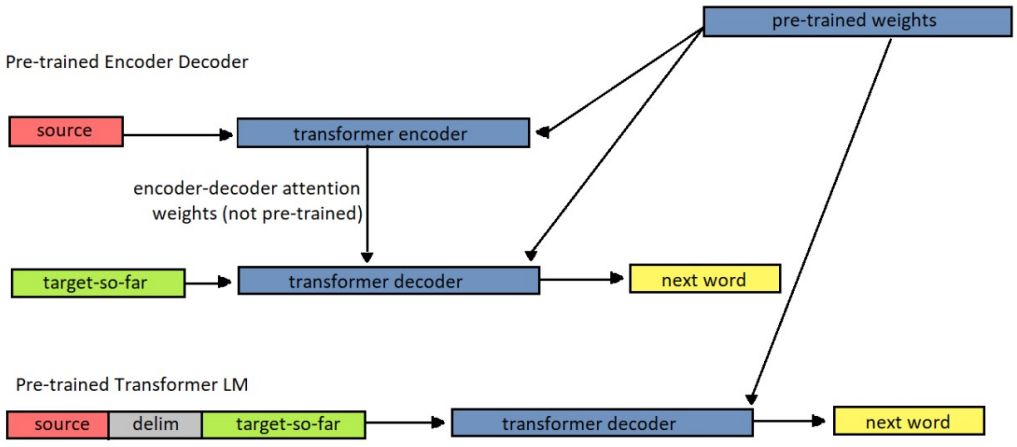

Figure 3. Text Summarization with Encoders and Decoders

14) stop

A. Function speaker recognition()

1) Import speakerverificationtoolkit.tools.
2) Define the path where samples are are saved.
3) Set the test file as Recording[i] and read it's path.
4) Extract the features of test file using extractmfc-cfromwavfile().
5) Compare the distance in features with the test file and each sample recording and find the sample which has the least distance.
6) Return the position of the sample found.

B. Function speechtotext()

1) Import pydub, speechrecognition and splitonsilence.
2) Set Recording[i] as the test file.
3) Using recognizer(), recognise the speech elements in function.
4) Use function adjustforambientnoise(), to remove the noise.
5) Split the file into chunks using splitonsilence() and pass the chunk to recognizegoogle().
6) Display each chunk.

C. Function textsummarization()

1) Import json, torch and t5 transformer.
2) Read the file temporary.txt and store it in a variable text after converting it into string.
3) Replace the characters such as '' and other stop words.
4) Pass the string to the encoder model generated using t5 transformer with parameters numbeams=4, norepeatngramsize=2, minlength=60, maxlength=534, earlystopping=False.
5) Pass the result of encoder to decoder to generate the final summary.
6) Display the summary

5. RESULT

The execution procedure for AMBOC is as follows: First, we created a data set which consists of sample voice recordings and meeting recordings of speakers. The dataset was preprocessed and converted to .wav file as shown in Figure 4. All the background noises were removed. These samples were then passed to our speech recognition module, where they were split to smaller chunks. This was done to split the silence parts of the speech as well as to improve accuracy. The speech was then converted to text format (.txt file) using Google speech to text API.

The voice samples were also passed to our speaker verification module where the identity of the speaker was verified using information from the sample recording and matching it with meeting recordings. Feature extraction was done using MFCC and then the voices were verified by computing the distance using DTW algorithm (Dynamic Time Warping.The output from Speech recognition was then passed to the Text summarization module to obtain summarized meeting minutes. Here we used T5 text-to-text transformer to obtain a summary from the text provided in input.

Our model, AMBOC was primarily designed to summarize English recordings. The extent of this research does not cover trying to summarize recordings in other languages but AMBOC can support many more languages with lower accuracy, as we are using Google API for speech to text and text summarization. We have compared the accuracy of the 3 main modules of AMBOC, with other existing modules and the results are displayed in the table 2 below:

The total accuracy that is obtained after combining all the models is resultant average accuracy of all the 3 machine learning models used, that is 83.33 percentage. AMBOC has the current highest accuracy and features compared to

Figure 4. Voice Recordings in .wav File

TABLE II. Comparison of Accuracy of Different Models

AMBOC MODELS	MODELS USED	ACCURACY(percentage)
Speech-to-text	HMM	73
	Google API	85
Speaker verification	Transfer learning	60
	MFCC and DTW	75
Text Summarization	TF-IDF	56
	Transformers	90

the other existing system for minute book creation using machine learning. It provides an end-to-end service for minute book creation. We have plotted a bar graph in Figure 5. based on the accuracy values obtained for each of models considered for AMBOC as shown below: A line graph is plotted for speaker recognition in Figure 6. representing mean accuracy and size of samples on each axis respectively. The graph was plotted using matplot library function. Our model was able to equally identify both male and female speakers with reasonable accuracy.

6. DISCUSSION

Meeting minutes, also called meeting summaries, are the written records of a meeting. They summarize and describe events in the meeting, and note down important agenda discussed and agreed upon by participants attending the meeting. By summarizing all the knowledge obtained by survey throughout and describe the first step toward the creation of automated minutes. The types of existing meetings, datasets and minutes displays an important disproportion between real meetings and the one that is used in the research. The data available are mainly of literature domain, but in the real-world scenario, it would be of business or topics of scientific domains. For these particular reasons, we decided to go beyond the literature domain and focus on other types of meetings as well, first of all on international online meetings. Our goal is to obtain summarized meeting minutes as it will be easier for the companies and offices to obtain feedback of their meetings and it will also allow them to save much time and money to keep an additional staff for the purpose of noting down meeting minutes. We make use of key technology like machine learning and speech-to-text transformation to derive key information from conferences and to make them instantly accessible. We consider meeting intention as one of the most appropriate scales in the multidimensional space of meeting types. We found out that business and decision-making meetings are most structured and require the most clear, structured and detailed agenda. The minutes obtained from such meetings will contain a list of decisions taken. For the meetings regarding information sharing and status updates, the agenda is also extremely important, here the minutes will be a refinement of ideas given in the agenda. For the linguistic form of minutes, it will be defined by the meetings themselves, as we will primarily use abstractive summarization methods. As for the datasets, we created a dataset that consists of both sample voice recordings and also meeting recordings are included. As most meetings, which are available to us, are held in English, we had chosen English as the main language for creating automated meeting minutes.

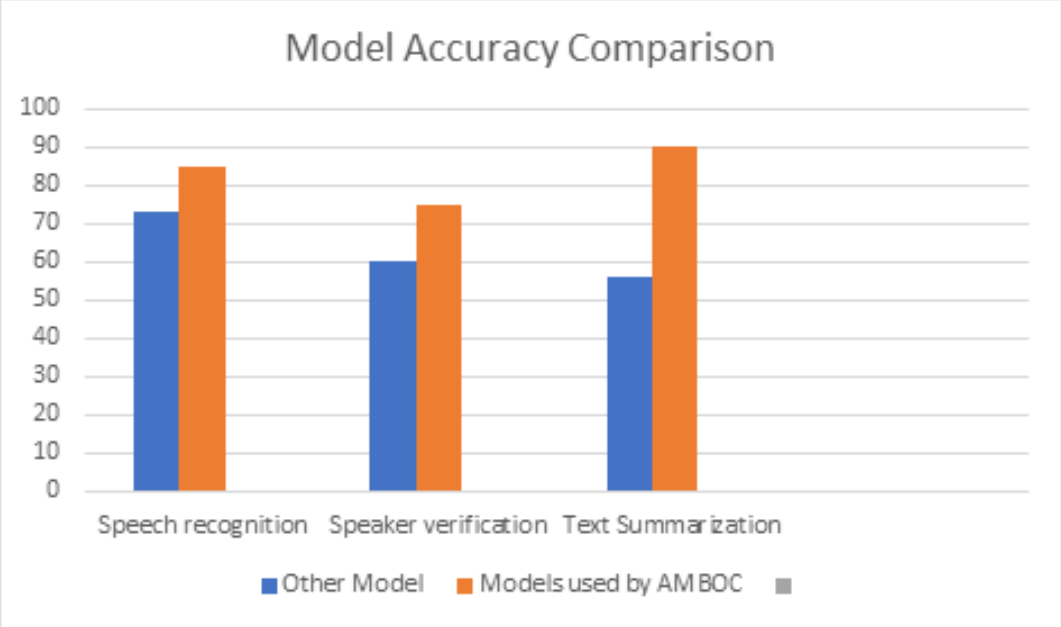

Figure 5. Model Accuracy Comparison Graph

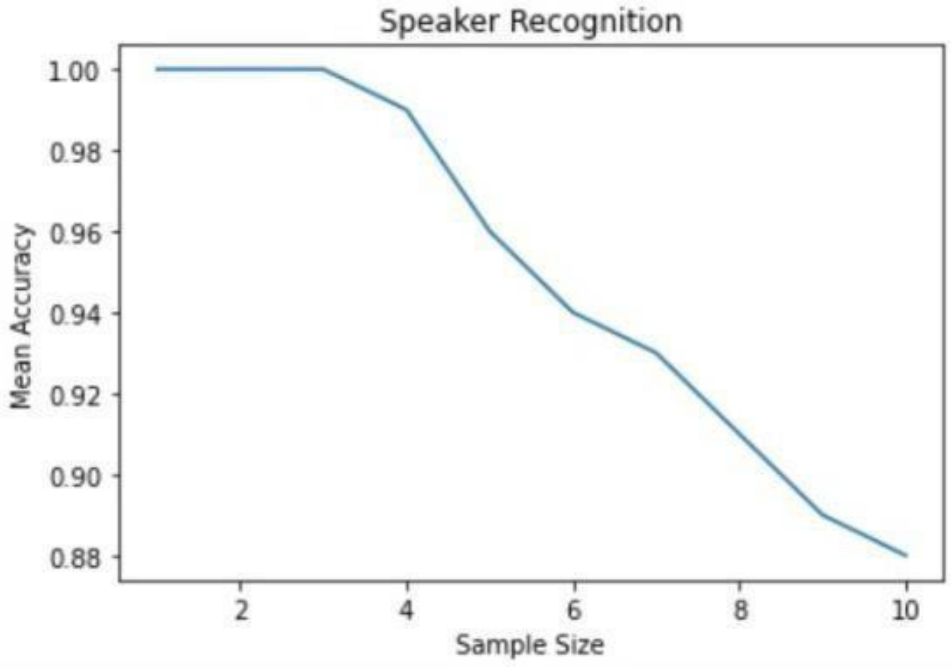

Figure 6. Line Graph of Speaker Verification

7. CONCLUSION

Meeting minutes is crucial for every meeting as it provides the documentation of the problem solving and gather ideas from each member in the meeting. The time has come for minutes of the meeting be automated for more efficient meetings. Automated meeting minutes has proved to reduce time consumption and cost of the organizations. Our study has presented the AMBOC as a well- organized technology which automate meeting minutes using deep learning techniques. We divided the experiment into three parts: speaker verification, speech recognition and text summarization correspondingly. A comparison study was made for each module. In the first module, voice features are extracted to form a unique voice signature and verification is done by comparing input voice samples to a list of enrolled voices. The speaker is recognized in two ways. One was Transfer Learning which achieved with an accuracy of 60 percent whereas MFCC and DTW achieved 75 percentage accuracy. To recognize the speech, the audio file was split in chunks based on silence in voice. Then the retrieved text can be passed by an HMM model or Google Speech API model. Speech recognition can be achieved by HMM or Google Speech API with accuracy of 73 percentage and 85 percentage respectively. The last module is text summarization. Text can be summarized in abstractive or extractive manner. Extractive text summarization was performed using TF-IDF model. This model achieved 56 percentage accuracy. We used Transformers for abstractive summarization. It achieved an accuracy of 90 percentage as shown in Table II.

To conclude, AMBOC is better in performance by using Google Speech API, MFCC and Transformers. This combination helps to provide productive and efficient meetings. This paper focused on upgradation of automation of meeting minutes. We have achieved better performance and usability compared to other papers mentioned in literature review. The paper closely related to our paper which produced minutes in Indonesian language has provided with state of the art accuracy. [5] Limitation of this paper is that only Indonesian language can be used. Our model can be used in various languages with lesser accuracy since Google API was used. With further experimentation, AMBOC model can show more improvement in the future.

REFERENCES

[1] Balaji V,G.Sadashivappa."MLLR Based Speaker Adaptation for Indian Accents",*International Journal of Computing and Digital Systems*, Vol 6, Sept 2017,pp. 293-301.

[2] Mohammed Usman."On the Performance Degradation of Speaker Recognition System due to Variation in Speech Characteristics Caused by Physiological Changes",*International Journal of Computing and Digital Systems*, Vol 6, May 2017,pp. 119-126.

[3] J. Wang, L. Lian, Y. Lin and J. Zhao, "VLSI Design for SVM-Based Speaker Verification System" in *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 23,July 2015,pp. 1355 - 1359.

[4] D.A. Reynolds, "An Overview of Automatic Speaker Recognition Technology", *IEEE International Conference on Acoustics Speech and Signal Processing*, vol. 4, 2002.

[5] G. H. Rachman and M. L. Khodra, "Automatic rhetorical sentence categorization on Indonesian meeting minutes," *2016 International Conference on Data and Software Engineering* (ICoDSE),2016.

[6] Zhang, Justin Jian Fung, Pascale Chan, Ricky "Automatic minute generation for parliamentary speech using conditional random fields. Acoustics, Speech, and Signal Processing". *ICASSP-88., 1988 International Conference on 10.1109/ICASSP* pp. 5536-5539.

[7] Athaya, Tasbiraha Munira, Sirajum Zaman, Afsana Zaman Hossain, Syed Kabir, Col. "A Proposed Algorithm and Architecture for Automated Meeting Scheduling and Document Management,2018.

[8] Beigi, H.," Fundamentals of speaker recognition".*Springer Science Business Media*,2011.

[9] Hossan, M. A., Memon, S., Gregory, M. A. (2010, December)." A novel approach for MFCC feature extraction". *In Signal Processing and Communication Systems (ICSPCS)*, IEEE,2010,pp.1-5.

[10] Zhang Wanli, Liang Guoxin."The research of feature extraction based on MFCC for speaker recognition,"International Conference on Computer Science and Network Technology, 2013.

[11] Leu, F. Y., Lin, G. L." An MFCC-based speaker identification system. In Advanced Information Networking and Applications (AINA)", IEEE,2017,pp. 1055-1062.

[12] Dumpala, S. H., Kopparapu, S. K." Improved speaker recognition system for stressed speech using deep neural networks". *In Neural Networks* (IJCNN), IEEE, 2017,pp. 1257-1264.

[13] Muda, L., Begam, M., Elamvazuthi, I."Voice recognition algorithms using mel frequency cepstral coefficient (MFCC) and dynamic time warping (DTW) techniques". arXiv:1003.4083,2010.

[14] Guangyu Kang "Variable sliding window DTW speech identification algorithm" IEEE,2009.

[15] S. Xihao and Y. Miyanaga, "Dynamic time warping for speech recognition with training part to reduce the computation," *International Symposium on Signals, Circuits and Systems* 2013.

[16] Dr.V.Ajantha Devi, Ms.V.Suganya,"An Analysis on Types of Speech Recognition and Algorithms", *International Journal of Computer Science Trends and Technology (IJCST)* Volume 4 ,2016.

[17] Claude Montacié, Marie-José Caraty, "A Si-

lence/Noise/Music/Speech splitting Algorithm",*ISCA Archive,5th International Conference on Spoken Language Processing*,1998.

[18] G.Saon,M. Picheny"Recent advances in conversational speech recognition using convolutional and recurrent neural networks".*IBM Journal of Research and Development*,DOI 10.1147/JRD.2017.

[19] Dominique Fohr, Odile Mella "New paradigm in Speech Recognition: Deep Neural Network" *IEEE Conference on Information System and Economic Intelligence*,2017.

[20] Ali Bou Nassif, Ismail Shahin, Imtinan Attili, Mohammad Azzeh, and Khaled Shaalan."Speech Recognition Using Deep Neural Networks:A Systematic Review".*IEEE ACCESS,DOI 10.1109/ACCESS.2019*.

[21] S. M. Mon and H. M. Tun, "Speech-to-text conversion (STT) system using hidden Markov model (HMM)," *International Journal of Scientific and Technology Research*, vol. 4, no. 6, Jun. 2015,pp. 349-352.

[22] Mbachu C,Akaneme S,Muoghalu C.N, "Filtering out noise in speech signal with fir filter based on single cosine term generalised adjustable window",*Indian journal of computer science and engineering*, Vol.10,2020.

[23] Song, Shengli and Huang, Haitao Ruan, Tongxiao. (2019). "Abstractive text summarization using LSTM-CNN based deep learning. Multimedia Tools and Applications". *78. 10.1007/s11042-018-5749-3*.

[24] Joachims,"A probabilistic analysis of the rocchio algorithm with TF-IDF for text categorization",cs cornell.edu1994.

[25] Cai T., Shen M., Peng H., Jiang L., Dai Q. (2019) "Improving Transformer with Sequential Context Representations for Abstractive Text Summarization." *Lecture Notes in Computer Science*, vol 11838. Springer, Cham,pp. 1-12.

[26] Maxime Peyrard; Beatriz Borges; Kristina Gligorić; Robert West., " Laughing Heads: Can Transformers Detect What Makes A Sentence Funny?", *arxiv cs.CL*, 2021.

[27] Arian Bakhtiarnia ; Qi Zhang; Alexandros Iosifidis., "Single-Layer Vision Transformers for More Accurate Early Exits with Less Overhead", arxivcs.CL, 2021.

[28] Alexander M. Rush,Sumit Chopra,Jason Weston."A neural attention model for abstractive sentence summarization."CoRR 2015.

[29] Khandelwal, Urvashi and Clark, Kevin and Jurafsky, Dan and Kaiser, Lukasz.Sample Efficient Text Summarization Using a Single Pre-Trained Transformer,2019.

[30] Sagarika Pattnaik,Ajit Kumar Nayak, "A simple and efficient text summarization model for odia text documents",*Indian journal of computer science and engineering*, Vol.11,2020,pp.825-834.

Megha Manuel has obtained her Bachelor's of Technology programme at Department of Information Technology, Rajagiri School of Engineering and Technology, Kerala affiliated to A.P.J Abdul Kalam Technological University. Her research interests are Machine Learning, Data Analytics, Software engineering and programming.She is currently working as a Data and Analytics Engineer.

Amritha S Menon has obtained her Bachelor's of Technology programme at Department of Information Technology, Rajagiri School of Engineering and Technology, Kerala affiliated to A.P.J Abdul Kalam Technological University. Her research interests are Data Mining, Machine Learning.She is currently working as a Software engineer.

Anna Kallivayalil has obtained her Bachelor's of Technology programme at Department of Information Technology, Rajagiri School of Engineering and Technology, Kerala affiliated to A.P.J Abdul Kalam Technological University. Her research interests are Audio Processing, Neural Networks and Cyber Security.She is currently working as a cyber security engineer.

Suzana Isaac has obtained her Bachelor's of Technology programme at Department of Information Technology, Rajagiri School of Engineering and Technology, Kerala affiliated to A.P.J Abdul Kalam Technological University. Her research interests are Machine Learning, cloud computing and Data Science. She is currently working as a full stack developer.

Dr.Lakshmi K.S obtained her B.Tech degree in Computer Science and Engineering from College of Engineering, Kidangoor. She pursued M.Tech in Computer and Information Science from Cochin University of Science and Technology, Kerala. She did her PhD program at SRM Institute of Science and Technology, Tamil Nadu, India. Under PhD program, she completed her research in the topic, 'A Novel Network Based method for disease comorbidity prediction' Her research interest includes Data Mining and Bioinformatics. Presently, she is also working as Assistant Professor in the Department of Information Technology, Rajagiri School of Engineering and Technology, Kochi affiliated to A.P.J Abdul Kalam Technological University.