MINUTES: HYBRID TEXT SUMMARIZER FOR ONLINE MEETINGS

Aaditya Mahadevan, Anina Pillai, Jigyassa Lamba

Department of Computer Science, KJSCE

a.mahadevan@somaiya.edu

anina.pillai@somaiya.edu

jigyassa.l@somaiya.edu

2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) | 979-8-3503-3509-5/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICCCNT56998.2023.10306385

***Abstract*— Online meetings have become commonplace, making it difficult to keep up. A summarizer for meetings, that captures the meeting content concisely, can prove to be beneficial. Meeting Summarization involves two key modules: Speech-to-text conversion that extracts transcripts from the meeting audio, and a hybrid text summarization model, incorporating extractive and abstractive summarization approaches. Extractive summarization was implemented using a centroid-based method that utilizes the compositional capabilities of word embeddings to capture the semantic relationships between sentences. The extracted sentences are fed to an abstractive model which uses the BART model as a general baseline. The BART model was fine-tuned on a dataset and optimized to generate better quality summaries. The T5 model has been used to generate one-line abstractive summaries. The models were evaluated using quantitative and qualitative testing. The summarizer thus assists the user to run through the important aspects of the meeting at a later stage.**

Keywords*— BART, T5, Abstractive Summarization, Speech-to-text, Extractive Summarization

I. INTRODUCTION

linguistics have led to substantial changes in how we interact with the world around us in recent years. The field of voice recognition is one of the most significant advancements. The complexity of language makes computational approaches increasingly difficult, and human communication is at the heart of breakthroughs in speech recognition. This is where natural language processing (NLP) approaches come into play. The use of natural language processing (NLP) opens up new avenues for improving human-computer interaction. Recent advances in the fields of Deep Learning and Artificial Intelligence have made human-level text interpretation more achievable. Deep Learning is now widely used in practically every facet of technology in the modern world. Thus with the aim of creating a Meeting Summarizer with the help of these state-of-art technologies, the application is being implemented. Extractive summarization is being implemented using a centroid-based method while the abstractive summarization part is being implemented using a BART model as a foundation; with T5 model being used to provide a single line summary that could serve as a title for the input data.

In this research paper, we propose an Online meet/video/audio/text summarizer, implemented in the form of a web application, that provides a concise summary of the input data in the form of a document that can act as minutes of the meeting. The use cases of this application include corporate meetings, single speaker lectures and conferences etc. Testing was carried out quantitatively (ROUGE and BLEU) as well as qualitatively (analysis by a domain expert) for the validity of the model.

Due to the advent of the internet and media in today's society, there has been an enormous increase in online gatherings, particularly as a result of the pandemic. Even while trying to pay full attention to the proceedings of the meeting, people usually seem to miss some key points of the meeting owing to background noise, connectivity issues, trying to take notes as the speaker progresses, or simply losing the flow of the meetings due to the natural need to multitask. In many use cases, it is also crucial to have a record and summarized version of the meeting for future needs. This is where meeting summarization becomes necessary.

Summarization is of two types: extractive and abstractive. Extractive summaries tend to be too identical to the source material, since sentences present in the summary are directly picked from the source, causing disruption in the flow of the text and cohesion. Thus, they lack a certain 'human touch'. Abstractive summarization uses deep learning techniques to generate, combine and omit sentences and words from the source, creating a summary that differs from the source text. Thus by introducing abstractive summarization into the picture, we can generate more human-like summaries. However, abstractive summaries may suffer with the problems of excluding details and key information from the source article. Hence, we assert a need for a hybrid model that encapsulates text without losing out on essential data. Advances in machine learning and computational

II. LITERATURE REVIEW

The two major components that constitute the project are the Automated Speech Recognition (ASR) module and the Summarization module. These modules necessitated substantial investigation, with important findings adapted to the project's flow.

The ASR module acts as the foundation of the project, since the summarized output is greatly dependent on accurate input from the user. Speech recognition is a complicated process that requires transforming speech data into word sequences using a function or set of rules. Speech to text conversion systems can be categorized into two: speaker dependent and speaker independent systems [1]. Speaker-dependent systems are concentrated on a single speaker, and therefore are more accurate for one speaker but less accurate

for others. Speaker independent systems, on the other hand, are modulated for multiple speakers.

Speech signals are difficult to interpret as they are randomly varying analogue signals. J.W. Picone [2] asserts that speech recognition systems involve two fundamental operations: signal modeling and pattern matching. Signal modeling is used to parameterize signals for representation. Feature extraction is a crucial step in signal modeling, involving the analysis of speech signals to obtain features like power, pitch, vocal tract configuration [3]. Recent trends have recognized the use of many feature extraction algorithms like Principal Component Analysis (PCA), Independent Component Analysis (ICA), Linear Discriminant Analysis (LDA), Cepstral Analysis and many more [4]. Many systems have been designed using these techniques.Su Myat Mon and Hla Myo Tun [5] propose a method for extracting characteristics from speech signals using Mel Frequency Cepstral Coefficients (MFCC), which are then fed into a Hidden Markov Model (HMM) to classify the spoken word. This method was limited to the use of solitary spoken words, such as flower, apple, and banana.

The second module of the proposed system is text summarization. This includes a hybrid approach to summarization i.e. the combined use of both extractive and abstractive approaches of summarization. Each approach was studied independently as well in combination with one another.

A novel method for extractive summarization called structured cosine similarity (SCS), provides document clustering with a new way of modeling on document summarization. It takes into account the structure of the documents in order to improve the quality, stability, and efficiency of document clustering [6].TF-IDF extracts data from numerous websites over the internet and provide an extracted summary of the information according to the user's query. The use of Selenium for web scraping is also described [7].Text Rank [8], a graph-based ranking system, uses recommendations to select the most essential sentences from the entire text: each text unit’s recommendation strength is determined by how many other units recommended it . It is ideally suited across fields, genres, and languages because it does not require any specific or domain knowledge. An unsupervised centroid-based technique for extractive summarization [9] proves to be more efficient than a single TF-IDF or Bag-of-Words model on multi-document and multi-lingual datasets. A condensed form of the document represented by the centroid, is made from addition of sentence vectors ranked by TF-IDF, and chosen according to a given threshold value. This centroid is projected into vector space along with the sentences in the document. The sentences closest to the centroid are chosen by calculating their cosine distance.

Sequence-to-sequence RNNs models are one of the most popular approaches of abstractive text summarization. It recommends encoding the entire sequence at once and then using that encoding as a context for generating the decoded or target sequence. A basic seq2seq model without attention, trained on paraphrasing tasks and then used for summarization is one such proposed method [10]. A hierarchical encoder is used to summarise the paragraph, so that summary can be generated. The fundamental disadvantage of this strategy is that the encoder has a hard time memorising lengthy sequences into a fixed length vector. RNN processing is sequential, which means we can't compute the value of the next time step until we receive the current time step's output. This makes RNN-based approaches slower.Abigail See, Peter J. Liu, and Christopher D. Manning suggested a hybrid pointer generator architecture with coverage in Pointer Generator Networks [11], and shown that it lowers inaccuracy and repetition. This approach allows for word copying via pointing as well as word generation from a fixed vocabulary. A generation probability $P_{gen} \in [0,1]$ is calculated for each decoder timestep, which weighs the likelihood of generating words from the vocabulary over copying words from the source document. It avoids attending to the same spots repeatedly when combined with coverage, and so avoids producing repetitive text.A novel inconsistency loss function is another way for ensuring that a unified model is mutually helpful to both extractive and abstractive summarization. Rather than relying solely on the complementary nature of sentence- and word-level attentions, it promoted these two levels of attention to be relatively consistent during training [12].The introduction of the transformer model in Google's research paper, Attention is all you need [13], was one of the most significant developments in abstractive summarization. The architecture is designed to handle long-range dependencies while solving sequence-to-sequence problems. Its fundamental goal was to completely handle input-output relationships with attention and recurrence. It uses multi-headed self-attention instead of the recurrent layers that are usually employed in encoder-decoder systems.BART (Bidirectional and Auto-Regressive Transformer) [14], which is pretrained on noise-corrupted text and a learned seq-2-seq model is created to reconstruct the original text, is advantageous as it provides a flexibility in taking even arbitrary lengths of shuffled text, lengthened by adding noise. The noising approach that gives the best performance is chosen. It is most useful when fine-tuned for text generation, but it also performs well in comprehension tasks.

To obtain the benefits of both extractive and abstractive summarization, an approach proposed the use of a combination of TF-IDF-TR (Term Frequency – Inverse Document Frequency – Text Rank) as an unsupervised learning algorithm and Seq2Seq (Sequence to Sequence) model as a supervised learning algorithm [15]. In a two-step process of extractive and abstractive summary, Transformer Language Models [16] proposed a way for summarizing large materials. The output of the extractive step is used to train the abstractive transformer language model. For text summarization, [17] presented a document level encoder based on BERT[Devlin et al. 2019. This model used interval segment embeddings and [CLS] tokens at the start of each sentence. On top of this encoder, various extractive, abstractive, and hybrid models were developed, compared, and fine-tuned with various types of optimizers.

Evaluation is a difficult task since there is no standard metric for summarization and human evaluators don’t have strong agreement across different types of summaries.For summary evaluation, it is important to point and choose which segments among the important points of the text should be preserved. Furthermore, the readability, coherence and factual correctness of the summary need to be kept in mind during evaluation. [18]. Traditional, human evaluation of summaries would prove to be very expensive and require extensive hours of manual labour. Hence the need for automatic evaluation rose, with cosine similarity, unit overlap and longest common subsequence being the earliest techniques to be proposed. But these evaluation metrics gave unsatisfactory results when compared to actual human like summaries. ROUGE (Recall-Oriented Understudy for Gisting Evaluation), one of the evaluation techniques was found be effective for summarization tasks and proved to have high correlation with human summaries in single document summarization [19]

While ROUGE has shown effectiveness in showing n-gram overlap between gold and generated summaries, it still isn’t a definitively effective solution to finding similarities between human summaries and system generated summaries, in part to its

inability to accurately identify synonymous discussions and content. ROUGE 2.0 tries to offset its predecessors disadvantages by capturing semantic overlap of such synonymous content and also provides topic coverage [20]

Taking into account all of the findings, we conclude that a hybrid approach for text summarization is required. Using solely an extractive model has a number of limitations, including the fact that it is only dependent on words and their frequencies, is ineffective at capturing document topic and context, and is incapable of handling synonyms. The study on a lone abstractive model, on the other hand, is currently ongoing. There is yet to be produced a flawless model that can mimic human-like summaries. Long document summaries, as well as maintaining the order of the summary in text comprising storylines and plots, are other key issues. Accuracy is the most important requirement for a text summarization system. By reading the summary, the user should understand the gist of the entire meeting or document. As a result, while summarising, there should be a balance between abstraction and extraction in order to construct a fluent, factually correct, and cohesive summary.

III. METHODOLOGY

The proposed hybrid summarizer incorporates both extractive and abstractive summarization techniques. The compositional capacities of word embeddings were used in the extractive model's implementation to capture the semantic interconnections between sentences. The abstractive BART model, which was built by fine-tuning the weights of the BART model on a pertinent dataset, uses the output of the extractive model as its input. In addition to text input, other media input types including audio, video, meetings, etc. were facilitated through the use of an ASR module. The system also contains a range of summary outputs, including solely extractive, hybrid, and one-line summaries that use the T5 model to describe the entire document in a single line. There is functionality to customize these outputs using various criteria, such as the word limit, the number of sentences in the output summary, and the percentage of information retention.

All the arrows depict the overall flow of the system with different operations depicted in Figure 1

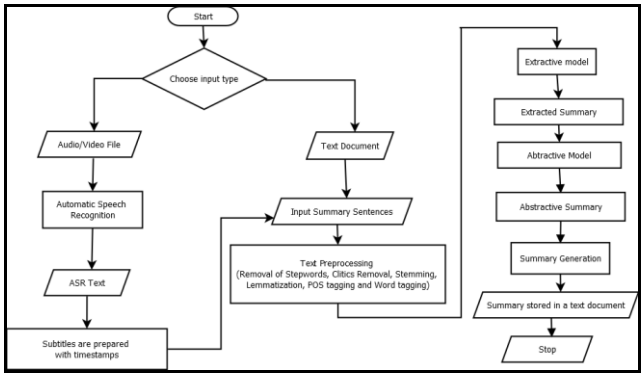


Fig. 1 Proposed System Flow

A. Data Input

The user will input data as audio, video (recordings from meetings), or text (pdf, doc, and txt files). When audio or video input is received, the audio is extracted and sent to the ASR module, which generates transcripts to reformat the input in the intended text format. Thereafter, the text feedback will be prepared in a way that the summarizer will recognize.

B. ASR Module

For the summarizer module to perform effectively, the ASR module, which provides input for the rest of the system to facilitate the summarization of meeting media files, must be accurate, comprehensive and error-free. High precision is therefore crucial for this module. The AssemblyAI API serves as the foundation for the ASR module's speech-to-text capability.

The API performs asynchronous transcription and generates JSON output, along with the transcripts that, after some formatting, serve as the input for the system. This output contains data about speakers, timestamps, confidence levels, punctuation, etc. Transcribing previously recorded audio or video files is known as asynchronous transcription. For lengthier meeting files, the computation time—which is roughly 15–30 percent of the file duration—can be laborious. Transcript generation, for instance, typically takes 90 seconds for a 10-minute clip, although it might occasionally take up to 3 minutes, in the worst case.

C. Text Pre-processing

The first step follows the common pipeline for every summarization task: preprocessing the input text. The document was broken into sentences using the tokenization functions provided by the nltk library. These tokenized sentences were further split into ino word tokens. All words were converted to lowercase. Tokens that do not contribute to the task at hand, such as stop words, special characters, and punctuation, were all removed, and the text was regularized. To allow the word embeddings to uncover the linguistic regularities of words with the same root, no stemming was done.

D. Extractive Model

When comparing closely related phrases, a bag of words or TF-IDF-based frequency-driven technique frequently fails to discern the semantic relationship between tokens. The suggested method provides an unsupervised technique to extract the most significant content, where it chooses the sentences that are most pertinent to the content of the source text using predetermined statistical and semantic approaches, rather than generating new phrases to build the summary. We tested a centroid-based approach for semantic analysis of a text that makes use of the configurational power of word embeddings. Additionally, based on the created intermediate representation, sentences are graded according to the average relevance of the terms in the sentence. A second step in the system pipeline is creating an extractive summary from the top k most crucial sentences, which is then input into the abstractive model.

1) *Embedding Model:* The summarizer employs word embeddings to choose phrases that, despite having distinct meanings from the most significant keywords or centroid words, have the same sense. A shallow neural network was deployed to learn complex semantic correlations using Word2Vec's simple vector field operators. To get the embedding model, the processed input text was parsed into word tokens and sent to Word2Vec.

2) *Generating Centroid Words:* Using the defined embedding model from the preprocessed tokenized word list and the TF-IDF-scoring algorithm, a centroid vector representing the most important words from the input text is created. The words with a TF-IDF score greater than the subject threshold would be the centroid words. The method locates the word centroid, aggregates the word vectors that make up the centroid, and produces the embedding vector representation of the centroid.

3) *Sentence Scoring:* The ranking and scoring of the sentences are based on how closely they resemble the centroid embedding vector. The technique derives the vector representation of each sentence in the preprocessed tokenized sentence list by adding up all the word vectors that make up a sentence and comparing them to the centroid embedding vector. Once the sentence embedding vectors have been established, the approach examines the correlation between the centroid and the sentence embedding utilizing vector similarity calculation measures such as cosine similarity, Euclidean distance, Jaccard index, etc..

4) *Defining Limit Parameters:* The sentences are sorted according to their similarity rankings in descending order. The top-ranked sentences are continually picked and included in the summary until the designated limit is reached. The limit can be of three types: word, sentence, and retention percentage. The word limit setting causes the system to select sentences from the sorted list until it reaches the predetermined limit, the sentence limit setting enables the model to select the top n sentences from the ranked representation, and the retention percentage limit setting calculates the number of sentences that fall within the predetermined percent to construct the extractive summary.

5) *Resolving Redundancy:* The issue of redundancy arises when there are too many similar sentences in the summary. By computing the cosine similarity between each sentence already in the summary and the incoming sentence at every iteration and rejecting the incoming sentence if the correlation value exceeds the predefined limit, the approach employs cosine similarity with a predefined threshold to meet the redundancy property.

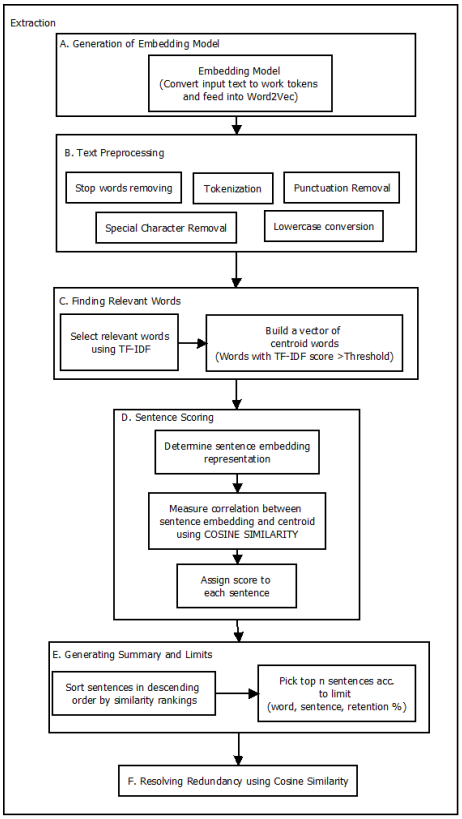


Fig. 2 Extractive Summarization Module

E. Abstractive Model

The abstractive model's input is derived from the extractive model's output. The abstractive summary generation model was fine-tuned on a relevant dataset using pre-trained BART weights. Additionally, one-line abstractive summaries that capture the essence of the text in a few words were generated using the Seq2Seq T5 model. For any NLP application, especially summarization, there is nearly always an advanced model that can be deployed as a starting point. To expedite training and improve performance, it is intended to import all of the weights from the pretrained neural network model and use them as a starting point. For the summarization work, we used a pre-trained model called "ainize/bart-base-cnn."

For the objective of acting as a baseline model, we assessed contemporary models such as "facebook/bart-large-cnn," "google/pegasus-multi-news," "t5/base," etc. However, it was determined that the 'ainize/bart-base-cnn' model weights were suitable for the task through manual testing on relevant topic data as well as testing via ROUGE and BLEU metrics. The performance of the models stated above can vary depending on the subject matter of the input text, but their results are practically all in the same ballpark. We discovered that the selected model produced coherent and accurate summaries more quickly than the others, though, after manually evaluating the models on inputs related to the target area. We chose "ainize/bart-base-cnn" as the base model for this task because it has a smaller model size than the other models we considered and because computation time is crucial for the use case.

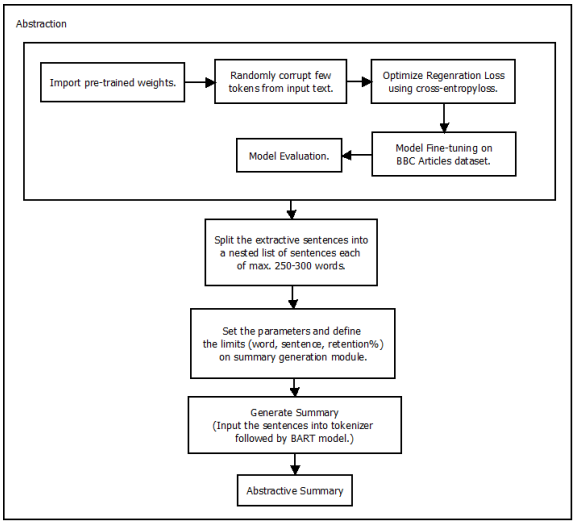


Fig. 3 Abstractive Summarization Module

1) *Datasets:* In order to train and fine-tune the model, two datasets were employed. The choice was determined after carefully weighing a number of variables, including the length of the input text, length of the gold summary, the topic relevance of the summary, the degree of abstraction, the addition of new phrases, the use of a wide vocabulary, etc. With well over 300,000 distinct news items written by CNN and Daily Mail correspondents, the CNN/DailyMail Dataset is an English-language database for summarization. The "facebook/bart-base" model was modified for the summarization function using the reference framework model, trained on this dataset. In addition, we used a publicly available dataset from Kaggle that included 2127 items from the BBC in a variety of news categories, including business, entertainment, politics, sport, and technology. It has two columns: "Summary",

which is made up of gold summaries; and "Text", which forms the input text. The original dataset's "highlights" were just a few brief gold summaries, which were quite inadequate for the requisite design. Therefore, we trained and modified the reference model weights using the BBC Articles dataset in order to better refine the model for the summarization of lengthier articles and to produce more in-context summaries.

2) *BART:* The Bidirectional and Auto-Regressive Transformer, also known as BART, is a sequence-to-sequence model that combines GPT and BERT, a bidirectional encoder and an autoregressive decoder. It uses a transformer-based Seq2Seq model to begin with the distorted source text and then attempt to denoise it by replicating the actual text from the decoder, with each decoder layer paying attention to the encoder's final hidden layer. Models for the "base" and "large" cases are two subtypes of pre-trained models. A small sample of the input/source text is randomly corrupted using noise techniques to prepare the model for pre-training. The regeneration loss is then optimised using the cross-entropy loss between the output and the decoder's output during training. Token masking, token deletion, token infilling, sentence permutations, and document rotation are some instances of noise transformation techniques.Furthermore, using BART to generate abstractive summaries from the input summary from the extractive model was substantiated by increases of up to 6 ROUGE scores in summarization tasks.

3) *Fine-tuning Weights:* To accommodate the BART model for the task of meeting summarization, the baseline model was modified using the BBC Articles dataset. The model was trained on the dataset for 50 epochs with an 8-sample batch size using the BART architecture. Only the most significant sentences produced by the extractive model made up the input length, which was set at about 300 tokens. Following that, ROUGE and BLEU measures were used to assess the refined hybrid model. Additionally, a comparison with contemporary models that are currently in use and a human evaluation were undertaken. Summaries ranging in length from 100 to 150 words were constructed using the hybrid algorithm.

4) *T5:* According to the Text-to-Text-Transfer-Transformer (T5) paradigm, all NLP tasks should be recast into a single text-to-text format with text strings serving as both the input and output. The T5 model is almost identical to the original Transformer model with the exception of removing the Layer Norm bias, shifting the layer normalization outside the residual path, and using a new position embedding technique. The T5 pre-trained model called "snrspeaks/t5-one-line-summary" was used to create one-line abstractive summaries. A T5 model was trained on 370k academic papers to provide a one-line summary based on the paper's abstract. It was trained using the PyTorch module for Python's simpleT5 library, which is designed for quickly training T5 models that use transformers and PyTorch. The three one-line summaries with the highest level of confidence can be used as the titles for the other two summaries.

F. Data Output

The output will be made available as a text document that can be downloaded in a number of different formats and contains a summary of the file that the user entered.

IV. IMPLEMENTATION

The aforementioned methodology was implemented as a web application. The website was developed with Django, a high-level Python web framework that allows for the rapid building of secure and scalable websites. AssemblyAI, a third-party API, was used to implement the ASR module.The Login module provides two different login mechanisms. The user can log in using their phone number and an OTP generated and validated using the Twilio API. Alternatively, the user can sign in to the web application using their Google accounts, which is facilitated by the Google-OAuth API. The extractive and abstractive summarization models are developed in Python and supported by packages such as nltk, scikit-learn, and gensim. Furthermore, libraries such as HuggingFace were employed for working with pre-trained weights, particularly for fine-tuning. The Word2Vec package was used to implement the word embedding model. The fine-tuning was performed with PyTorch, and the training was conducted with an NVIDIA Quadro RTX 5000.

V. TESTING & EVALUATION

It can be challenging to determine a summary's level of quality. A variety of different metrics can be used to compare how well summarization systems work. A system summary can be compared to the original text, a human-generated summary, or another system summary. Although human evaluation appears to be the logical course of action, it is typically not practical due to its slowness and expense, and the results from various individuals are not always comparable. Thus, automatic evaluation is the more typical and accepted approach. Using ROUGE and BLEU Scores, we are undertaking intrinsic evaluation in this project.

1) *Quantitative Results:* To automatically evaluate summaries, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [Lin 04] is widely adapted as it tends to correlate well with human judgments. The following metrics are utilised to report the outcomes of this thesis: ROUGE-1 stands for Unigram Overlap, ROUGE-2 for Bigram Overlap, ROUGE-L for Longest Common Subsequence, and ROUGE-Lsum for Union LCS. Recall, Precision, and F1 Score are some of the ROUGE measures that can be calculated. Additionally, BLEU (BiLingual Evaluation Understudy) is used as a metric to assess text that is given as output from the machine, by comparing the machine-translated text with high quality reference translations. The BLEU score gives a value that is between 0 and 1.We obtained the ROUGE scores using the pyrouge package. We tested a number of alternative approaches to text summary while keeping as many variables as possible constant. The results are tabulated below:

TABLE I

ROUGE F1 SCORES ON GIVEN INPUT TEXT

Models	ROUGE		
	ROUGE-1	ROUGE-2	ROUGE-L
Pegasus	48.6	26.1	25.9
T5	37.7	28.6	25.0
Text Rank[a]	47.6	26.8	28.1
Ours(Hybrid)	51.9	30.2	33.9

[a] Extractive Model. All ROUGE scores have a 95% confidence interval of at most ±0.25 as reported by the official ROUGE script.

TABLE II

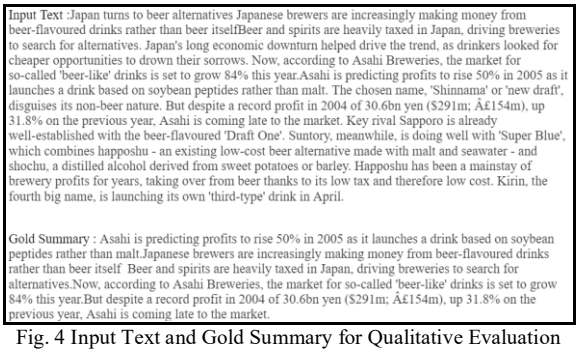
BLEU SCORES

Models	BLEU			
	BLEU-1	BLEU-2	BLEU-3	BLEU-4
Pegasus	31.4	17.1	13.2	11.3
T5	20.0	14.3	11.3	8.6
Text Rank[a]	33.1	19.6	17.5	16.5
Ours(Hybrid)	30.6	22.6	20.2	18.7

[a] Extractive Model.

Inferences: The hybrid model incorporates additional generated sentences as well as sentences directly extracted from the input text. Many tokens directly from the text are included in the gold summary, which accounts for the hybrid model's slightly exaggerated ROUGE scores. In comparison to Pegasus and T5, the hybrid model's summary length is on average longer, contributing to a high ROUGE metric. We can conclude from the following statistics that our model has at least comparable performance when compared to current contemporary models. For very concise summaries such as T5, ROUGE-1 alone may suffice, especially as we are also applying stemming and stop word removal. This accounts for the particularly low scores for Pegasus and T5 as the number of n-grams increase. However, the hybrid model performs the best in case of ROUGE-L scores. Unigram recall and precision count all co-occurring words regardless their orders; while ROUGE-L counts only in-sequence co-occurrences. LCS does not require consecutive matches but in-sequence matches that reflect sentence level word order as n-grams, it automatically includes longest in-sequence common n-grams, therefore no predefined n-gram length is necessary. Thus, a higher ROUGE-L score for the hybrid model proves that this model captures sentence level structure in a natural way. The model was also tested against BLEU scores which assess how closely the generated summary matches a human expert's reference summary. Individual N-gram scores are calculated by evaluating just matching grams in a certain order, such as single words (1 gram) or word pairs (2-gram or bigram). Even with growing n-grams, the hybrid model fared comparably well when evaluated using BLEU scores to the Pegasus model, which showed excellent results.

2) *Qualitative Results:* Using HuggingFace library, we performed a qualitative analysis for comparing the summaries generated by various other summarization models.


Input Text :Japan turns to beer alternatives Japanese brewers are increasingly making money from beer-flavoured drinks rather than beer itselfBeer and spirits are heavily taxed in Japan, driving breweries to search for alternatives. Japan's long economic downturn helped drive the trend, as drinkers looked for cheaper opportunities to drown their sorrows. Now, according to Asahi Breweries, the market for so-called 'beer-like' drinks is set to grow 84% this year.Asahi is predicting profits to rise 50% in 2005 as it launches a drink based on soybean peptides rather than malt. The chosen name, 'Shinnama' or 'new draft', disguises its non-beer nature. But despite a record profit in 2004 of 30.6bn yen ($291m; Â£154m), up 31.8% on the previous year, Asahi is coming late to the market. Key rival Sapporo is already well-established with the beer-flavoured 'Draft One'. Suntory, meanwhile, is doing well with 'Super Blue', which combines happoshu - an existing low-cost beer alternative made with malt and seawater - and shochu, a distilled alcohol derived from sweet potatoes or barley. Happoshu has been a mainstay of brewery profits for years, taking over from beer thanks to its low tax and therefore low cost. Kirin, the fourth big name, is launching its own 'third-type' drink in April.

Gold Summary : Asahi is predicting profits to rise 50% in 2005 as it launches a drink based on soybean peptides rather than malt.Japanese brewers are increasingly making money from beer-flavoured drinks rather than beer itself Beer and spirits are heavily taxed in Japan, driving breweries to search for alternatives.Now, according to Asahi Breweries, the market for so-called 'beer-like' drinks is set to grow 84% this year.But despite a record profit in 2004 of 30.6bn yen ($291m; Â£154m), up 31.8% on the previous year, Asahi is coming late to the market.

Fig. 4 Input Text and Gold Summary for Qualitative Evaluation

Model Used	Summary	Inferences
Text-Rank	"Asahi is predicting profits to rise 50% in 2005 as it launches a drink based on soybean peptides rather than malt.Japanese brewers are increasingly making money from beer-flavoured drinks rather than beer itself Beer and spirits are heavily taxed in Japan, driving breweries to search for alternatives.Now, according to Asahi Breweries, the market for so-called 'beer-like' drinks is set to grow 84% this year.But despite a record profit in 2004 of 30.6bn yen ($291m; Â£154m), up 31.8% on the previous year, Asahi is coming late to the market."	-> Extractive Model -> Matches with the gold summary. -> Sequence is not maintained, which is essential in case of stories or plots.
Pegasus	"Now, according to Asahi Breweries, the market for so-called 'beer-like' drinks is set to grow 84% this year.Asahi is predicting profits to rise 50% in 2005 as it launches a drink based on soybean peptides rather than malt."	-> Abstractive Model -> Though the model is abstractive, the summary generated is purely extractive in nature.No new words are formed. -> Nothing can be derived after reading the summary.
t5 Base	"market for 'beer-like' drinks is set to grow 84% this year . despite record profit in 2004 of 30.6bn yen ($291m; £154m), asahi is coming late to the market . beer and spirits are heavily taxed in japan, driving breweries to look for alternatives . but despite a 31.8% increase on the previous year, the company is predicting profits to rise 50% in 2005 as it launches a drink based .. .s ."	-> Abstractive Model -> Limit : 512 tokens -> Extra tokens at the ending of summary. -> Incoherent sentences (Highlighted in red)
Our Model	"Asahi Breweries predicts profits to rise 50% in 2005. 'Shinnama' or 'new draft' disguises its non-beer nature. Suntory is doing well with 'Super Blue', which combines happoshu and shochu. Happoshu has been mainstay of brewery profits for years, taking over from beer thanks to low tax and therefore low cost. Asahi predicts profits will rise by 50% as it launches a drink based on soybean peptides. The chosen name disguises the nature of the beer, disguising it as non-Beer. It is hoped that the beer-flavoured drink will appeal to consumers."	-> Hybrid Model (The output from the extractive model is fed to the BART based abstractive summarizer as input.) -> No limit for token length since we are first generating important sentences through an extractive model. -> Associating proper nouns and changing tenses (Highlighted in green) -> Removing irrelevant information (Highlighted in blue) -> New lines are added according to the context of the input.(Highlighted in Yellow)

Fig. 5 Qualitative Analysis of Various models with inferences

VI. CONCLUSION

We finally complete the analysis after comprehending and completing the project's implementation and testing using various assessment methodologies. Minutes is a web-based summarizer system that generates a brief and specific outline of meeting recordings or text documents comprising the proceedings in the form of a downloadable text file. A novel hybrid technique to text summary was developed, which incorporates the benefits of both extractive and abstractive summarization approaches while partially ameliorating the limitations produced by a solitary abstractive or extractive model. By integrating abstractive aspects and creating additional relevant information, the hybrid model mitigates the issue of maximum token constraints and loss of critical data while preserving the 'humanness' of the summary. To find the most salient ideas in the text and include key information, the extractive model uses a token centroid-based method that makes use of frequency-driven concepts like TF-IDF, mathematical concepts like cosine distance, and strategies that take advantage of the semantic relationship between the tokens, like embedding models.The BART model's weights served as the foundation for the abstractive summarizer. This model was utilized to produce abstractive summaries, and it was then fine-tuned on a dataset to fulfill the specifications for summarising online meeting recordings, such as longer input text lengths, speech discrepancies, and subject dependence. A model built using the T5 architecture weights was also used to generate one-line abstractive summaries. The self-contained ASR module was then integrated with the summarizer to support audio and video media files.

REFERENCES

[1] NishantAllawadi,'Speech-to-Text System for Phonebook Automation', Computer Science And Engineering Department Thapar University, June 2012.

[2] J. W. Picone, "Signal modelling technique in speech recognition," Proc. Of the IEEE, vol. 81, no.9, pp. 1215-1247, Sep. 1993.

[3] Manish P. Kesarkar, Feature Extraction for Speech Recognition, M.Tech. Credit Seminar Report, Electronic Systems Group, EE. Dept, IIT Bombay, Submitted November2003

[4] Sanjivani S.Bhabad, An overview of technical progress in speech recognition, International Journal of advanced research in computer science and software Engineering, Volume 3, Issue 3, March 2013

[5] Su Myat Mon, Hla Myo Tun, Speech-To-Text Conversion (STT) System Using Hidden Markov Model (HMM), International Journal of Scientific & Technology Research Volume 4, Issue 06, June 2015

[6] Soe-Tsyr Yuan and Jerry Sun, Ontology-Based Structured Cosine Similarity in Document Summarization: With Applications to Mobile Audio-Based Knowledge Management, IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)

[7] K Usha Manjari, Syed Rousha, Dasi Sumanth, Dr. J Sirisha Devi, Extractive Text Summarization from Web pages using Selenium and TF-IDF algorithm, Proceedings of the Fourth International Conference on Trends in Electronics and Informatics (ICOEI 2020)

[8] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing Order into Text. In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

[9] Rossiello, Gaetano, Pierpaolo Basile, and Giovanni Semeraro. "Centroid-based text summarization through compositionality of word embeddings." Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres. 2017.

[10] Chi Zhang, Shagan Sah, Thang Nguyen, Dheeraj Peri, Alexander Loui, Carl Salvaggio, Raymond Ptucha, Semantic Sentence Embeddings for Paraphrasing and Text Summarization.

[11] Abigail See, Peter J. Liu ,Christopher D. Manning, Get To The Point: Summarization with Pointer-Generator Networks, Association for Computational Linguistics (2017).

[12] Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, Min Sun, A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss, Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics.

[13] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin, Attention Is All You Need, Advances in Neural Information Processing Systems 30 (NIPS 2017).

[14] Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, Luke Zettlemoyer, BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension, Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.

[15] Meena S M, Ramkumar M P, Asmitha R E, and Emil Selvan G SR, Text Summarization Using Text Frequency Ranking Sentence Prediction , 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP)

[16] Sandeep Subramanian, Raymond Li1, Jonathan Pilault, Christopher Pal, On Extractive and Abstractive Neural Document Summarization with Transformer Language Models, Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP).

[17] Liu, Y., & Lapata, M. (2019). Text summarization with pretrained encoders.

[18] Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., & Kochut, K. (2017). Text summarization techniques: a brief survey. arXiv preprint arXiv:1707.02268.

[19] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." In Text summarization branches out, pp. 74-81. 2004.

[20] Ganesan, Kavita. "Rouge 2.0: Updated and improved measures for evaluation of summarization tasks." arXiv preprint arXiv:1803.01937 (2018).