Business Meeting Summary Generation using Natural Language Processing (NLP)

MSc Research Project
Data Analytics

Srishti Subhash Chandra Prasad
Student ID: x20142218

School of Computing
National College of Ireland

Supervisor: Majid Latifi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Srishti Subhash Chandra Prasad
Student ID:	x20142218
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Majid Latifi
Submission Due Date:	31/01/2021
Project Title:	Business Meeting Summary Generation using Natural Language Processing (NLP)
Word Count:	8877
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	31st January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Business Meeting Summary Generation using Natural Language Processing (NLP)

Srishti Subhash Chandra Prasad
x20142218

Abstract

Text summarization is a challenging task in the domain of NLP (Natural Language Processing). In this modern period, where enormous amounts of data are available online, it is challenging to provide a better model for extracting information efficiently and quickly. Manually extracting the summary of a huge written document is quite challenging for humans. Researchers used to focus mostly on extractive ways, but there's been a steady shift in the stream of research toward abstractive ways as it is more difficult to implement. Meeting summaries compress the most important things spoken at a meeting while maintaining the meeting's original meaning, as reading through the complete transcripts is time consuming and costly to the company. The dataset was taken from the ICSI corpus which contains 75 meetings, lasting around 72 hours. The meetings have an average of 6 attendees, and each transcript contains an average of 1000 lines to process for summarization. The meeting summarization is achieved by combining both the abstractive and extractive approaches: the extractive approach incorporates feature extraction based on TextRank or graph-based ranking algorithm and word frequency, while the abstractive approach uses deep learning technique which comprises of RNN, LSTM with an attention mechanism model . The performance or summary evaluation was calculated using ROUGE and human evaluation.

1 Introduction

Text summarization is the process of constructing a concise, cohesive, and better summary of a lengthier text document, which includes highlighting the text's important points. The goal is to develop a clear and concise summary that incorporates document's major concepts and concentrates on the most significant details while keeping the overall meaning. This reduces the time it takes to comprehend extensive documents such as meeting transcripts while without deleting crucial information. Many researchers have yet to investigate the topic of summarizing conversations or gatherings. Text summarization is used in various domains such as analysis of legal documents, social media marketing, medical cases, newsletters, question answering bots, science and R&D. Summarization is clearly crucial for a meeting recording repository, as it will benefit users in finding appropriate meetings and locating key segments of recordings for viewing. Reading summaries saves a lot of time compared to listening to or watching a recording. It's also not a good idea to look up or read the original transcripts because they contain a lot of untrusted channels, explanations, and side topics that don't contribute anything to the substance (Buist et al.; 2004). While summarizing meetings may appear to be similar like summarizing broadcast news,news articles or customer reviews,in theory, it appears to be quite different in practice. In contrast to written texts, where sentence limits are easy to recognize, talks involve a variety of vocal inflections. Meeting information is also substantially less than news articles or broadcast, both of which are essentially highly compressed information.

Text summarization is a popular and challenging area of study in NLP (Andhale and Bewoor; 2016; Widyassari et al.; 2020). Its used to evaluate long texts by constructing summaries from raw input text. Employees devotes a huge amount of time to attend the meetings as meetings are an important part of their lives as they used to discuss ideas, share knowledge, and make plans/goals. This might help participants utilize time and be more effective at work. Meeting are expensive and unnecessary in general, and they waste both time and money. In actuality, the majority of meetings end with no good understanding of what has been discussed or, more significantly, what's been decided. Meetings consume a massive amount of an employee's worktime, with various studies suggesting that 50% of a meeting's time has been consumed on without any productive issues and 25% on irrelevant topics. Despite the fact that 69% of employees say meetings are useless and pointless, it is hard to avoid or miss any meeting at the workplace (Romano and Nunamaker; 2001; Allen et al.; 2012).

Summarizing is challenging for a variety of reasons, including the lack of fixed rules for evaluating whether a summary is appropriate or not, or which is more clear and to the point. There are no specific scores or assessments by which the generated summaries may be judged. It's difficult to evaluate if that text is faultless or objectively excellent. Collecting proper training data is usually difficult, costly, and insufficient. The subjective nature of a human summary evaluation involves judgments such as conciseness, readability, manner, understandability, clarity, and completeness (Hahn and Mani; 2000). There is no score that is both acceptable to human judgment and simple to evaluate at this period. Employees can identify key details in the meeting transcription with the use of a meeting summary. Individualizing a summary is a basic technique that needs a long time to complete. As a outcome, automated summary production is becoming increasingly common as a way of getting basic understanding of long texts, and deciding what information is significant in a given input document is a difficult task. Two of the most major obstacles in text summarizing are the difficulty in identifying the most relevant information from an original input text, as well as the difficulty in presenting that crucial information in the resulting summary.

This research paper examines the below following research questions:

RQ1: How can a combined extractive and abstractive techniques will help to create a human-readable and short summary from a lengthy business transcripts using natural language processing that can save human efforts and time?

RQ2: Is it possible for the abstractive technique to comprehend the meaning of words from an original non-structured input transcript and provide a non-repetitive and short meeting summary as output?

Following are the objectives of research in order to answer above research questions:

- The research objectives is to study the related work in the domain of text summaraization with its different approaches (abstractive and extractive).

- Proposing a research methodology to perform text summarization and machine learning algorithm using deep neural networks with python and ML libraries such as nltk, numpy, tenserflow, keras using LSTM and RNN.

• Implementing machine learning algorithm with natural language processing which is essential in the field of summarizing to develop Automatic Text Summarization (ATS) model.

• Evaluation of the developed model on different metrics such as ROUGE scores and human evaluation.

The novelty of the research paper is that it proposes a model with the combination of Abstractive and Extractive Text Summarization which combines the merits of both the approaches to produce less redundant, concise, short, readable and understandable summaries from the multiple long business meeting transcripts that will help the employees as well as organization to save their time and human efforts by implementing TextRank, RNN, LSTM with attention mechanism in which researcher rarely worked on ICSI dataset for business applications using hybrid approach, dealing with multiple files.

In this research project, the main contribution or achievement is to select an appropriate model from various models used earlier in an extractive and abstractive approach and merging it according to the business meetings applications. There are numerous methods used in extractive as well as abstractive approaches previously, but to consider which model is more suitable from all those approaches or techniques and combining it appropriately to have better results in terms of business meeting summaries because all prior study was focused on their size of the input, languages, output nature, areas, topics, and summary method. Using the ICSI corpus dataset, it combined TextRank, RNN, LSTM, and attention mechanisms in extractive and abstractive approaches, focusing primarily on the business meeting summarizing application. It takes multiple meeting transcript files and generates legible output summaries with a higher ROUGE metrics and human based analysis, that will help to handle large multiple transcript files in the future, mostly for business meeting applications.

The following is how the remaining part of the paper is organised: In Section 2, it discuss the related work in the field of abstractive and extractive text summarization with different techniques. In Section 3 it explains the methodology that is utilized to construct the project. In Section 4, it describes the workflow of design and the approaches. In Section 5 it outlines the proposed implementation of the research project. In Section 6 the evaluation of the results with various experiments is performed. In Section 7 the paper is accomplished with conclusions and ideas for future work.

2 Related Work

Due to the abundance of data available today, automatic text summarization has become crucial for obtaining the proper amount of content in less time from massive texts. All of the previous review articles examined used a variety of methodologies and approaches to construct a concise summary from various sorts of documents, each with its own set of virtues, flaws, and future potential. With the aid of Google API, Balasundaram and Amalraj (2019) turns the recorded audio into text (speech recognition). This paper's main objective is to summarize the speaker's voice document. It utilized a hybrid approach, combining abstractive and extractive techniques. NLP has been explored for many years in a variety of fields due to its use. The review papers are typically split into Extractive and Abstractive techniques, which are logically separated into parts which give the result of Rouge-1 as 39.9, ROUGE-2 as 18.1 and ROUGE-L as 35.6.

2.1 Extractive Summarization Using Various Methodologies

As per Shirwandkar and Kulkarni (2018), the extractive text summarization technique employing deep learning was applied to a single document. It employs a mix of RBM (Restricted Boltzmann Machine) and fuzzy logic to preserve the text's original meaning while ensuring that no information in the output summary is degraded. The information provided is in English. Differing elements, such as distinct phrases with various degrees of words in them, were used to provide a relevant summary. Using RBM and Fuzzy logic, it created 2 summaries per document. After that, using various processes, the summaries are reviewed and integrated to give a final summary. Fuzzy logic was used to improve the accuracy of the summary RBM (unsupervised learning algorithm), which resulted in F measures of 84% and 88% precision. This model produced far better results than the prior strategy, which relied solely on RBM. Text overloading problem is overcomes in this. This work may be expanded in the future by using it on many papers and in multiple languages, and by combining alternative methodologies and other elements for a much improved summary. Abstract summarization will be used in future studies.

Madhuri and Ganesh Kumar (2019) used a statistical strategy to create extractive text summarization based on the rating of the sentence extracted from a particular input text. The study was limited to a single document. The phrases were first allocated weights based on their significance, and then their weights were used to rank them. The most highly scored sentences are used in the summarizing, resulting in a summary of high-quality. Using Python and NLTK, the system was evaluated on five papers with a total of 20 sentences. The sentence that was more than rank 8 is created as an output summary, according to their study. When compared to previous approaches, this model is more accurate. The next step was to try out additional texts and improve the accuracy.

Kaur and Srivastava (2019) proposed an improved and revised Partial Textual Entailment (PTE) method based on graph for one file using an extractive approach in order to overcome the drawbacks of the prior research, which stated that Textual Entailment (TE) represents the relation between the lines from interconnectivity and node signifies sentences in the graphs with certain restrictions. They used the PTE technique rather than TE on several datasets and found superior results when compared to current algorithms. As the summary demands additional text inputs and diverse document clusters, the future potential here was to implement the same technique and perks of the PTE techniques to multi files.

JUGRAN et al. (2021) solves the problem of the paper, which relied on the NLTK library to parse the raw text input line by line. In order to reduce the time spent creating the summary, the SpaCy library is used instead of NLTK, which is a superior alternative in relation to time savings. As SpaCy provides an object-oriented approach, this method turns the entire text into an object. It links the word for word prediction that was previously unavailable in the program. Because it requires extra training in better understanding the entire content at once, the future work is employing abstractive summarization to utiliz RNN and LSTM as future approaches. It even plans to expand their study by using a hybrid technique that marries extractive and abstractive summarization.

For legislative sessions, Zhang (2012) employed structural summarization. Rhetorical modelling is a tough approach in comprehending the extractive summary of audio transcripts. Prior research has found that identifiers in paragraphs, such as styles, typefaces, titles, subtitles, and delimiters, play a crucial role in summarization. It suggested a Conditional Random classifier, a one step strategy that was superior to the two steps Rhetorical structure and assessing major phrases approach previously employed. For N-gram and acoustic characteristic, the findings were 68 percent and 66.7 percent ROUGE-L, F-measures, respectively. Future work in this research is to involve using this technique in far more interactive meetings and training the CRF classifier in one-step for better comprehension of meetings in respect of chunks, which will help improve the performance.

From a single input text, the author of a paper done by Haider et al. (2020) suggested the K-Means clustering technique. On the input file, it employed Gensim word2vec, that was able to obtain semantic subjects in a systematic manner. That was the result of combining K-Means with Genism. The dataset utilized in this study came from a BBC news report. Sentences were graded based on the presence of nouns and numeric data, and the numerical method resulted in higher results for the articles of business, as this technique prioritizes numerical data. The same procedure may be used in the future to extract information from several papers, yielding a better result.

By retrieving summary elements, Bokaetf et al. (2015) targeted a multi-party meeting. The researcher of this work ran a test on the previously run algorithm to evaluate the results on the retrieved elements once more. It provides a new procedure that is superior to the one that was earlier used. The AMI meeting corpus provided the data for this research. The suggested technique was developed as an unsupervised algorithm that separated conference transcripts in smaller parts that reflect a meeting event. They also intended to improve the pre-existing keyword extraction summary algorithm, which may be utilized to raise the standard of summarization.

The confusion network (CN) and the n-best hypothesis are two types of structures introduced in this research (Xie and Liu; 2011) for increasing the quality of meeting summarization. For employing diverse words and phrases, it employed an unsupervised approach. It can boost the effectiveness of result using 1 best recognition by utilizing ROUGE 1, 2 metrics on the data of ICSI corpus. Human transcripts are close to rich speech recognition transcripts (RSR). To address the restriction of the earlier study, it looked at the mistake rate of a word retrieved from a sentence summary. If the goal is to choose segments, RSR can perform as well or superior than human transcription. Re-scoring the probable words on its extraction summary, that might enhance the result by employing CN, is planned for the future. Using a supervised framework with n-best and CN hypotheses, the same strategy may be used.

According to Liu et al. (2011), they utilized a supervised technique for keyword extraction and even a single-loop feedback mechanism to promote a link between the summaries and extraction of keywords. The dataset was taken from the ICSI conference corpus and included a human-written transcript. In all of the trials, it outperforms the unsupervised TF-IDF. It produced a positive outcome while analyzing the n-best hypothesis. Because of the frequency oriented method, challenges in assessment, human supervision, and a high failure rate in speech processing, several problems develop in keyword extraction, leaving room for further research.

Because it is feasible for one or more phrases to share similar knowledge, Gunawan et al. (2019) offered the way to eliminate the shortcomings of repetition. As a result, the emphasis was mostly on reducing redundancy caused by several publications. It included a number of different web news stories. Pre-processing the combination of articles was the first step. Second, utilizing relationship measurement within them, the TextRank technique was used to extract the important information. Finally, the MMR (Maximal Marginal Relevance) method was used to reduce the similarities. ROUGE 1's F-score was 0.5103, and ROUGE 2's F-score was 0.4257, as a result of the preceding procedures. The first letter and incorrect words are two limitations of this suggested model that lead to lower accuracy.

The suggested article by Merchant and Pande (2018) demonstrates the application of the text summary technique to legal text since it is often critical for lawyer to read long judgments in a short amount of time. It implements the LSA (latent semantic analysis) NLP approach with a single document. The data for this is gathered from official government websites. For various documents and single documents, trained and untrained techniques were utilized. ROUGE 1 gave it a score of 0.58 on average. The disadvantage of this work was that it used LSA in the middle, which broke the continuity. The long-term goal is to strengthen the system's ability to overcome obstacles, as well as to integrate it on a mobile platform to boost usability.

Padmanandam et al. (2021) presented an interactive visualization model in which a user may send commands and comprehend the interpretation of a large number of articles. They developed a word cloud, which illustrates the frequency of a keyword by making it bigger or smaller. When the keyword is larger, it is more commonly used; whenever a keyword is smaller, it is less commonly used. On the proposed method, this can be seen. The occurrence of words seen in the source text had a big factor. The constraint is it's only available in English. Future development in this area included different languages support as well as reading and analyzing real-time updates utilizing patterns and popular posts.

This section covers all of the papers that were examined using the extractive method. It is separated into extractive and integrative since both methodologies must be understood in order to combine and conduct additional activities, according to the study topic. Because it pulls out significant sentences from the input, the extractive technique was more linguistically precise and simple to use. The numerous algorithms used in this method resulted in distinct sorts of summary scores. The majority of the papers focused on a single document and used techniques such as fuzzy, NLTK library, PTE, LSA and, K-means. The correctness of the summaries was determined using TF-IDF or ROUGE metrics. The abstractive strategy employed for merging with extractive will be discussed in the next part, which will aid the plan in achieving extended advantages.

2.2 Abstractive Summarization Using Various Methodologies

The semantics and structural based method of summary creation from the raw text (Rahul et al.; 2020) was the subject of this study. It utilised a variety of datasets, both single and multiple, including DUC2000 and CNN datasets. It looked at prior work, its flaws, benefits, and potential future scope. This suggested methodology led to the finding that all of the summaries created were unique, with some being inadequate or unnecessary. ROUGE and TF-IDF scores are utilized to get a more accurate and short summary. Because there is no precise understanding of what the good summary is, the suggested model's future scope was to improve the existing model so it can deliver higher accuracy.

The approach was proposed by Badgujar et al. (2018), which was based on a graph and used abstractive summarization. It primarily focused on the emotion infusion approach, where a graph node contains numerous characteristics such as tags, word counts inside the text, and sentence and word positions. It estimated the shortest route, that was restricted to a single document in this case. To address this flaw, an abstractive parser based on semantic data was employed, which may be utilized on many texts.

An abstractive model was developed based on seq2seq with additional characteristics on a single document was proposed in the publication (Hao et al.; 2020). The Daily Mail and CNN provided the dataset for this study. It used 2 types of models for capturing network features that were better than the old way and produced better results. With R-1 at 5.6 percent, R-2 at 5.3 percent, and R-L at 6.2 percent, it was a much more effective and updated version. It will deploy its methodology with extractive summarization in the future to provide both benefits.

The fundamental goal of the sequence-to-sequence model of RNN, based on the article (Mohammad Masum et al.; 2019), was to develop a legible, intelligible, efficient, and quick abstractive summary. It made use of an Amazon dataset including Kaggle food reviews. In the encoding and decoding layers, LSTM and RNN (bidirectional) with attention models were implemented. The calculation of missing words, embedding of words, language processing, and vocabulary were all challenges in this suggested approach. It primarily focused on lowering train loss while boosting accuracy, resulting in a more abstractive approach than before, as well as achieving the desired outcome with a 0.036 value. The paper's drawback was that it provided a better summary for short texts but not for big texts. It was said that in the future, it will focus on creating good summaries regardless of whether the content was short or long, and that there would be no set length.

This section covers the abstractive analysis of the work. There had been fewer studies in the abstractive method than in the extractive approach because the abstractive technique is more complicated because it develops its own lines from raw input. Types of documents were both multiple and single type, and the algorithms were also primarily RNN seq2seq or graph based, although the majority of the paper's limitations were restricted to short text. Long text proved tough since an abstractive technique scans the entire material at once and creates a summary. If there are more than 100 lines, it is difficult to construct a summary, even for humans. CNN/DailyMail, kaggle, or a manual written document created from speech-to-text provided the dataset. ROUGE score and frequently human evaluations were used in the evaluation.

3 Proposed Research Methodology

This section explains the method that is utilized to build the project. From processing the input meeting transcript to receiving the summary output, there are several processes involved in producing short and accurate summaries. Meeting transcripts come in a variety of formats, depending on the topic, the number of participants, and whether the meeting was lengthy or short. It might even have background noise, jargons or any other type of disruption in the meeting, all of which can lead to incorrect output. Natural language processing (NLP), that is essential in the field of summarization, is used in this proposal.

Figure 1 depicts the well-defined tasks that must be completed in order to get the right intended output by following the processes outlined below in accordance with the meeting application. Because the length of the dialogues in the meeting summaries varies depending on the topic and speakers involved, the meeting summaries can be long or short.

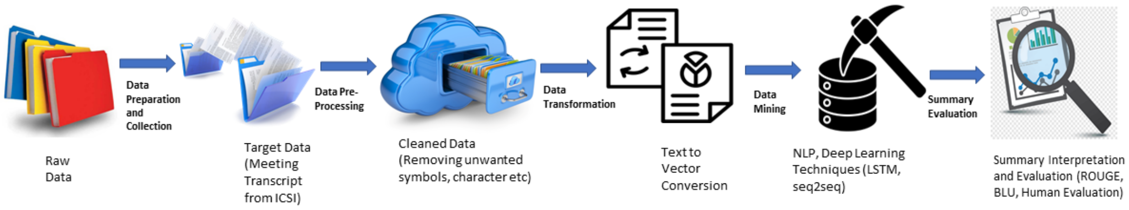

Raw Data
Data Preparation and Collection
Target Data (Meeting Transcript from ICSI)
Data Pre-Processing
Cleaned Data (Removing unwanted symbols, character etc)
Data Transformation
Text to Vector Conversion
Data Mining
NLP, Deep Learning Techniques (LSTM, seq2seq)
Summary Evaluation
Summary Interpretation and Evaluation (ROUGE, BLU, Human Evaluation)

Figure 1: Proposed Research Methodology

3.1 Data Collection and Preparation

The first stage is to collect relevant meeting transcripts in order to process them for the research and achieve the research objectives. The data was extracted from the ICSI corpus [1], which included 75 meetings that lasted around 72 hours. Each meeting has an average of six people in attendance, and each transcript has over 1000 lines to analyze for summarizing. With the aid of the NITE XML toolkit, the annotations or transcripts were stored in nxt format. The meetings include various date-and-time stamps, tags, participant information, notes, and channels. The datasets are in zip format and are organized by signals, media, or audio. They must be correctly processed before proceeding since they contain background noise, crosstalk, and must be converted from.mrt to .txt format. The following steps is to examine the meeting area and type. It's similar to assessing the meeting transcripts in the beginning to gain a general understanding of what actions should be followed based on its nature, and then taking action or pre-processing procedures in the following stage to determine if the summary generated is correct or not. After determining the meeting's domain, the following stage is to determine if the meeting documents are single or multiple, based on the subjects covered by the number of attendees.

3.2 Data Pre-processing

After the data has been gathered and before using a summary technique, pre-processing is an essential step. The meeting transcripts gathered may contain jargon or background noise, resulting in poor summaries. As a result, data must be pre-processed or cleaned before it can be analyzed directly. To improve the summary from the dataset, words that are repeated, signs, special characters, sentences that may be unfinished, incorrect words and phrases are removed from the transcript. Because the information was captured in its natural state, it contains a significant portion of noise in background and crosstalk, which is cleaned during the pre processing stage. According to the participants, utterances, and stop words, the meeting transcripts are processed and cleaned. Because the transcripts were so long and contained several files, they needed to be condensed into fewer lines and processed with 60 transcripts containing 100 lines, which were then saved in a folder for further steps.

[1] https://groups.inf.ed.ac.uk/ami/icsi/

3.3 Data Transformation

The dataset must be translated into a vector before an operation can be performed on it to extract short summaries. Text vectorization in NLP methodology is a key sentence or set for evaluating words by mapping them with multiple ways. Vectorization can be used to identify words or phrases. For data transformation, ML libraries as scikit-learn are employed. The TF-IDF was used to identify the most important terms in the transcript and provide scores based on their value in the transcript. The cleaned text is utilized to extract relevant characteristics and convert them into vectors that the NLP algorithms used to select the key sentences to include in the summary. It employs the word frequency features in combination with the Term Frequency-Inverse Document Frequency (TF-IDF) technique that's basically use for data extraction and summarization.

3.4 Developing Summarization Algorithm using NLP and Deep Learning Techniques

Abstractive and extractive summarization are two techniques to text summarization. The abstractive method is the major focus in this stage, with certain features of an extractive approach incorporated. Some techniques of deep learning, such as sequence to sequence and long short-term memory (LSTM), are combined with NLP. I employed the TextRank algorithm, that is a graph-based NLP ranking method based on Google's PageRank algorithm and developed models utilizing LSTM and RNN using scikit-learn, tensorflow, keras and deep neural network, machine learning and python libraries. For NLP, RNN is a popular neural network design. For creating language models and voice recognition problems, it has shown it is quite fast and accurate. However, while RNN identified dependencies and would only learn from recent data because it understands context and current dependencies, LSTM helped in solving this challenge. As a result, LSTMs are a unique type of RNN in which knowing context will be helpful.

3.5 Summary Evaluation

The model has been built and must be validated and trained. For testing and training the model, the input text and summary (csv) dataset is split into 70:30 ratio. The optimizer, epochs, batch size, loss function, learning rate and embedding dimension are all hyper parameters that are tuned during training the model. The testing and training accuracy and loss are calculated in every sample and presented in a graph to provide clearer insights of the process through which the model is correctly trained and produces less loss and higher accuracy. As per the literature review, there are no specific or predefined scores that would tell anything beyond that score. The result is assessed using either a ROUGE score or a human review. The summaries should be brief, short, intelligible, and readable without compromising the overall meaning, as required by the aim.

As a summary, methodology was broadly categorized into two sections. The first section discusses the several stages of actions that were undertaken on the input transcript in order to generate a possible output summary. The second section explains how to examine using ROUGE metrics for evaluating summary of extractive approach utilizing TextRank and merged with the abstractive approach (LSTM, RNN, attention mechanism) in addition to increase accuracy and overcome the limits encountered in previous research.

4 Design Specification

The design and flow of the Abstractive and Extractive models are discussed in this section. It also explains the multiple parts that helped to build the model.

The merging of Abstractive and Extractive Summarization model is proposed in this research, and it consists of two major steps: Extractive stage and Abstractive stage. The first stage's output is taken as the second stage's input. The work flow of the summarization model is shown in Figure 2

4.1 Stage 1: Extractive Summarization

The collection of the input data is the first stage in our summarizing process. Meeting transcripts from the ICSI Corpus were used to create the dataset. When the input is ready, it is fed into the summarizing model, which predicts the input text's summary. The following are the three major phases as shown in Figure 2:

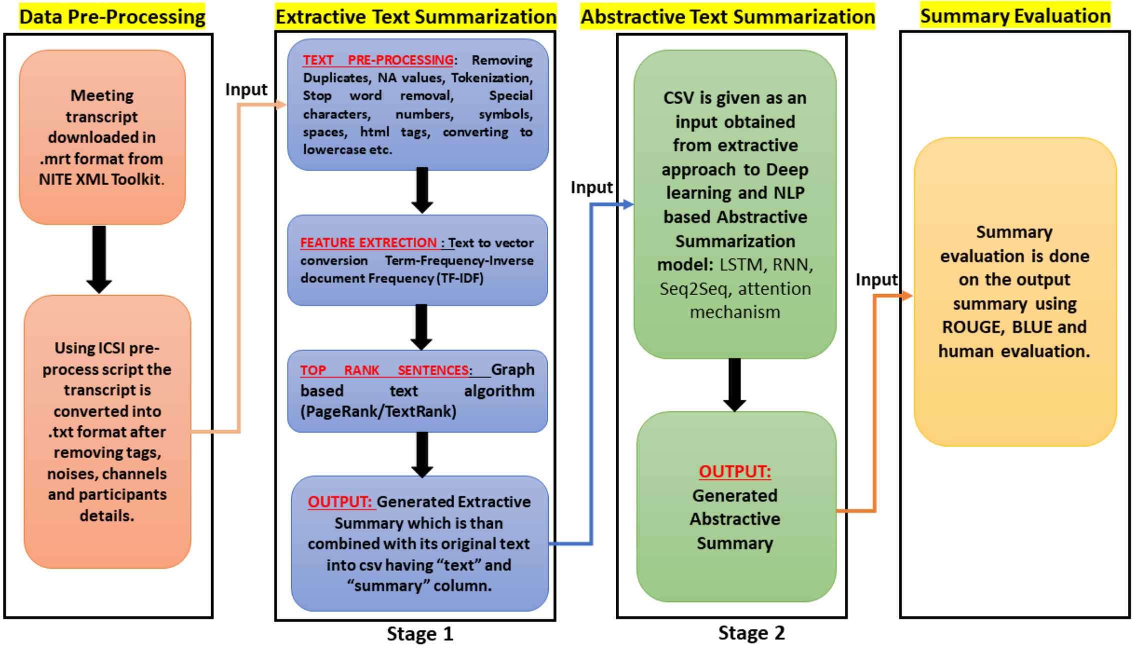

Data Pre-Processing
Meeting transcript downloaded in .mrt format from NITE XML Toolkit.
Using ICSI pre-process script the transcript is converted into .txt format after removing tags, noises, channels and participants details.
Input
Extractive Text Summarization
TEXT PRE-PROCESSING: Removing Duplicates, NA values, Tokenization, Stop word removal, Special characters, numbers, symbols, spaces, html tags, converting to lowercase etc.
FEATURE EXTRECTION : Text to vector conversion Term-Frequency-Inverse document Frequency (TF-IDF)
TOP RANK SENTENCES: Graph based text algorithm (PageRank/TextRank)
OUTPUT: Generated Extractive Summary which is than combined with its original text into csv having "text" and "summary" column.
Stage 1
Input
Abstractive Text Summarization
CSV is given as an input obtained from extractive approach to Deep learning and NLP based Abstractive Summarization model: LSTM, RNN, Seq2Seq, attention mechanism
OUTPUT: Generated Abstractive Summary
Stage 2
Input
Summary Evaluation
Summary evaluation is done on the output summary using ROUGE, BLUE and human evaluation.

Figure 2: Model Workflow

4.1.1 Transcript Pre-processing

The original input data contains a lot of useless, meaningless, not structured and irregular data that will be difficult for systems to analyze and handle. Transcripts are downloaded from ICSI corpus and were in the form of .mrt format which is converted to the .txt format using "icsi_preprocessing.py" script and stored the data into a folder. After that, the cleaned transcripts were reduced to small sizes by minimizing the lines and reducing files from 75 to 60 files because of the computational limitation. To improve the summary from the dataset, repetitive words, participants information and channels, NA values, HTML content, signs/symbols, incorrect words, stop words and sentences, special characters, sentences that may be unfinished, are removed from the business transcript and also eliminated during the pre-processing stage as it was recorded naturally in the meeting rooms.

4.1.2 Feature Extraction

For this phase, the processed transcript is fed into the extractive summarization model, which extracts important elements and converts them into vectors that the NLP algorithms used to select the key sentences to include in the summary. It include the Term Frequency-Inverse Document Frequency (TF-IDF) approach in combination with the word frequency feature, which is generally used for information retrieval and summarization. It's used to calculate scores of sentences based on the scores of TF-IDF words within these sentences. The number of times specific words appear in a transcript is described as Term Frequency (TF). If the meeting transcripts contain a large number of transcripts, each one with a distinct length, then the TF divides it by the total count of words in the text. IDF is determined by total number of transcripts divide by the number of transcripts that include that word.

4.1.3 PageRank/TextRank and Extractive Summary

The vectorized formatting of the lines or sentences are converted into a graphical representation in this phase using the TextRank algorithm, that is taken from Google's PageRank algorithm. It is based on graph technique for ranking. The connections linking the nodes reflect the matching score between the lines, while the nodes themselves indicate the sentences. The topmost sentences are taken and combined to produce the final summary, which is in the form of an extractive, depending on a specified threshold value. After generating the summaries with the extractive approach, it is merged with all the original text to create a csv with 2 columns named ”text” and ”summary” which contains texts of 60 transcripts and their generated extractive summaries with it. That is given to the abstractive approach as an input to produce the summary.

4.2 Stage 2: Abstractive Summarization

The abstractive approach uses the summary generated with extractive from stage 1 as input in step 2, which is in the form of csv combined with its original text. In this step, it uses neural networks to implement a deep learning approach. Recurrent Neural Network and Convolutional Neural Network are two forms of neural networks that are commonly used. CNNs are recommended in tasks like facial recognition and image classification in which the input length is fixed, and the result of each stage is entirely dependent on the input of the current state. As a result, the Encoder Decoder Sequence-to-Sequence design is ideal for this task. It employs RNN for decoder and encoder of both the networks as it uses records within the internal storage from all past outputs and current input to estimate the output of the current state. RNNs, on the contrary, are just useful for extremely short sequence, although Long Short Term Memory is an improved form of RNN which can help with long-term dependence problems. In addition, it used the Attention mechanism to forecast the output if sequences were extended by focusing just on specific areas of the input section. As a result, RNN LSTM Sequence-to-Sequence Decoder-Encoder with attention mechanism is used in phase two of the text summarization method. The encoder receives the full sequence of input and encode it in a single fixed-size context vector while the decoder then decode to create the abstractive summary as an output.

5 Implementation

This section covered the system configuration, as well as the various tools, libraries and software that were utilized to implement the research model. With full description of the dataset that was used for the business meeting generation.

5.1 System Configuration

The development of Abstractive and Extractive summarization approach used Python 3.7. Python was chosen as it is simple to use, extensively used for machine learning and natural language processing tasks, has a large number of libraries which can be easy to import and has a large online communities and councils to support it. I used a variety of IDEs, including Spyder, jupyter, PyCharm and local system, to test the ideas. I opted to utilize Google Colab because of the machine's bad performance and hangup difficulties and it helps the executions go quicker. It's simple to develop and run the code because it doesn't demand any installation on the local system, and the transcript preprocessing was done in Visual Studio. Up to a certain computing capability, colab provides both TPU and GPU hardware accelerators for no charge with Google free tier service. While the code is being executed, it might modify the runtime type. The free GPU is a 2496 CUDA cores with 1xTesla K80, while the TPU is 8 cores with a TPU v2 and about RAM of 12GigaBytes having RAM of 36GigaBytes as the maximum limit. It utilized the TPU accelerator since it is quicker than the GPU.

5.2 Dataset Description

The dataset for the meeting summarization is taken from ICSI (International Computer Science Institute) Corpus [2], which is natural recordings taken place at the ICSI. The annotations are saved in nxt format with the help of NITE XML toolkit. It includes 75 meetings that were recorded over a lengthy period of time (70-100 hours) in various formats, including audio and video recordings with transcripts. Each meeting has an average of six people in attendance, and each transcript has over 1000 lines to analyze for summarizing. The meetings contain different date-time stamps, tags, information about participants, notes and channels. The datasets are in zip format and are organized by signals, media, or audio. They must be correctly processed before proceeding since they contain background noise, crosstalk and must be converted from .mrt to .txt format. Transcripts are lengthy and includes channels, participants details with dialogues between the participants on some specific topic with the time stamps in it.

5.3 Model Implementation

Firstly, the original meeting transcripts are extracted from ICSI corpus. Then, from the extracted summaries obtained from the extractive approach, it is merged with the original text, parsed the various attributes and transformed into a CSV file by importing pandas python library. The task of text summarizing was performed using the 'text' and 'summary' fields. The dataset CSV file was imported into DataFrames with the help of pandas library, and the two required attributes were extracted and utilised for further

[2] https://groups.inf.ed.ac.uk/ami/icsi/

processing. It examined at 60 meetings as a sample transcripts having 100 lines for text pre-processing because of computational issues.

It used the sent tokenize modules using NLTK library and removed repetitive and null values from the input transcripts. NLTK was used to eliminate HTML content, stop words, symbols or numbers, special characters and excessive spaces. After the data has been cleaned, the further step is to extract key features from it. The text input was transformed into vector representations using the TF-IDF weighting algorithm. Importing the module of feature extraction from the scikit-learn python ML library accomplishes this. The vector representation is turned into a graph based on the networkx python library's PageRank method. After that, depending on the similarity metrics between the lines, the topmost lines are retrieved and given to the extractive summary as stage one's output. This outcome, together with the original text and extractive summary, is imported into a dataframe, that is then turned into a CSV file for further stage.

The abstractive text summarization model is created in the 2nd step with the help of the keras and TensorFlow ML and Neural Networks python libraries. To assist the model identify whenever the sequence begins and finishes, append the tokens for START and END as 'sostok' and 'eostok' respectively to the extractive summary. The dataset is divided into two sets 70:30. An embedding layer for decoder and encoder networks, as well as an attention layer to memorize extended sequences, make up the model, which is a three-layer LSTM encoder and a one-layer LSTM decoder, and a function of SoftMax activation to the output layer. The embedding layers are 200 units while hidden layers are 300 units in size and the hidden layer has 0.4 value as a dropout to minimize overfitting and increase performance of the model. The model.fit and model.compile keras modeling functional methods are used to execute for training the model when it has been built. The loss values and accuracy are calculated and evaluated by setting several Hyper-parameters like batch size, optimizer, number of epochs, embedding dim, learning rate, loss function, and activation. Following the training stage, the further stage was the inference stage, where it feeds the testing data into the model and receives the projected summary as a result.

Numerous experiments were carried out through training of the model, including Hyper-parameter tuning and analysis of the model's expected summary.

6 Evaluation and Results

In this section, I addressed various experiments carried out to generate a proper and good summary from multiple long meeting transcripts and the evaluation of the summaries using ROUGE, and human evaluations. It is important to evaluate the summaries to get a better idea of improvement in future steps.

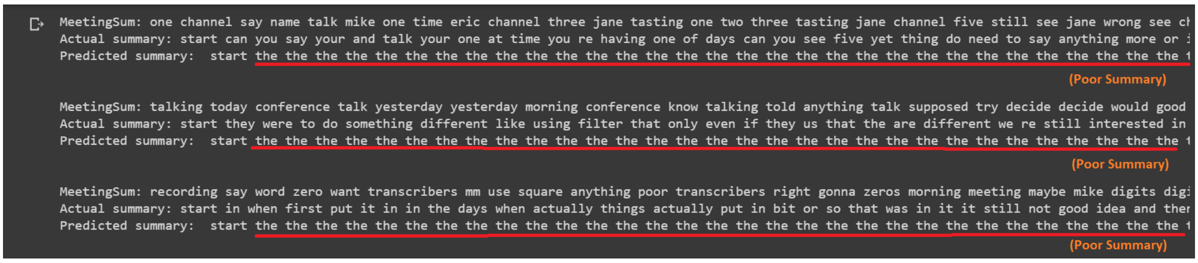
6.1 General Experiments on Input Transcripts

Here the experiments were performed on the input transcripts by reducing the lines in the transcripts files, reducing the number of files and reducing the summary lines for it. Firstly, the 75 transcripts were passed with more than 1000 lines as it is to the model, and generated 10 line summaries from it. After merging both 'text' and 'summary' the CSV size was too large from which it was crashing the RAM because of limited memory. So, decided to reduce the number of transcripts, number of lines in a transcript and also the generated summary size. Secondly, the 60 transcripts were passed to the model with

50 lines of text in it and 5 lines of summaries which did not give the proper output because to train a model it also requires a good amount of data as an input. Finally, the experiments were carried on using 60 transcripts, 100 lines of text with 5 lines of summaries merged into CSV and given to the abstractive model. Splitting data, epoch number, learning rate, batch size, embedded and latent dim and selecting length of text and summaries to pass were analyzed.

6.2 Experiment 1

The experiment 1 was performed using 70:30 training and testing dataset. It used latent and embedded dim as 256 for both. The number of epochs is 50 and the batch size to 128. Maximum text length and maximum summaries kept as 400 and 200 respectively. The output as shown in Figure 3 showed poor summaries. It was having lots of repeated words in it, shown with red lines in which one word was repeating most of the time.

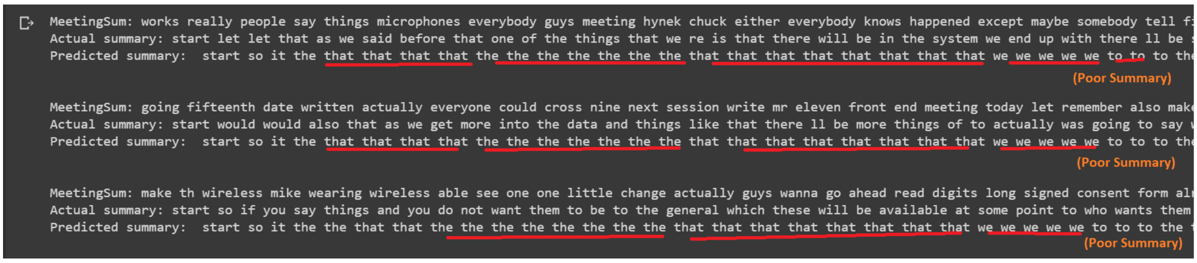


```
MeetingSum: one channel say name talk mike one time eric channel three jane tasting one two three tasting jane channel five still see jane wrong see ch
Actual summary: start can you say your and talk your one at time you re having one of days can you see five yet thing do need to say anything more or i
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                        (Poor Summary)

MeetingSum: talking today conference talk yesterday yesterday morning conference know talking told anything talk supposed try decide decide would good
Actual summary: start they were to do something different like using filter that only even if they us that the are different we re still interested in
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                        (Poor Summary)

MeetingSum: recording say word zero want transcribers mm use square anything poor transcribers right gonna zeros morning meeting maybe mike digits digi
Actual summary: start in when first put it in in the days when actually things actually put in bit or so that was in it it still not good idea and ther
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                        (Poor Summary)
```

Figure 3: Experiment 1

6.3 Experiment 2

The experiment 2 was performed using 90:10 training and testing dataset. It used latent and embedded dim as 256 for both. Using multiple epochs and batch sizes was fitted in this model in which first level epoch: 20, batch size: 64, secondly, epoch: 31 batch sizes: 64 and lastly, with epoch: 100 , batch size: 64. Maximum text length and maximum summaries kept as 460 and 220 respectively. The output as shown in Figure 4 showed poor summaries. It had repeated words, but multiple words came in the summaries, which was better than the previous one. It showed with red lines in which multiple words repeating multiple times.

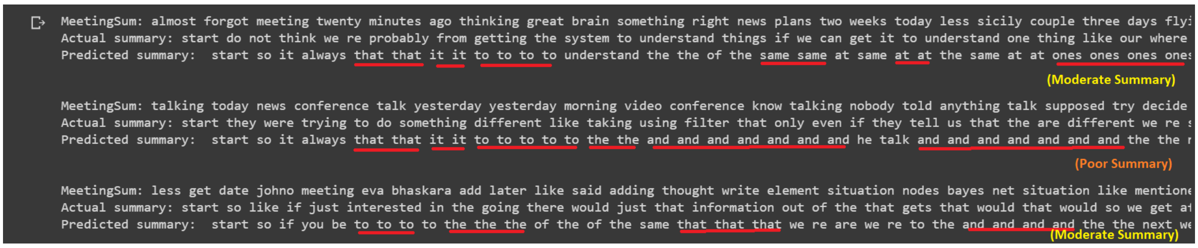


```
MeetingSum: works really people say things microphones everybody guys meeting hynek chuck either everybody knows happened except maybe somebody tell f
Actual summary: start let let that as we said before that one of the things that we re is that there will be in the system we end up with there ll be s
Predicted summary:  start so it the that that that that the the the the the the the that that that that that that that that we we we we we to to to the
                                                                                                                        (Poor Summary)

MeetingSum: going fifteenth date written actually everyone could cross nine next session write mr eleven front end meeting today let remember also make
Actual summary: start would would also that as we get more into the data and things like that there ll be more things of to actually was going to say w
Predicted summary:  start so it the that that that that the the the the the the the that that that that that that that that we we we we we to to to the
                                                                                                                        (Poor Summary)

MeetingSum: make th wireless mike wearing wireless able see one one little change actually guys wanna go ahead read digits long signed consent form al
Actual summary: start so if you say things and you do not want them to be to the general which these will be available at some point to who wants them
Predicted summary:  start so it the the that that the the the the the the the the that that that that that that that that we we we we we to to to the t
                                                                                                                        (Poor Summary)
```

Figure 4: Experiment 2

6.4 Experiment 3

The experiment 3 was performed using 80:20 training and testing dataset. It used latent dim as 300 and embedded dim as 200. The number of epochs is 50 and the batch size to 128. Maximum text length and maximum summaries kept as 470 and 220 respectively. The output as shown in Figure 5 showed poor and moderate summaries. It was having fewer repeated words than the previous experiment, shown with red lines.

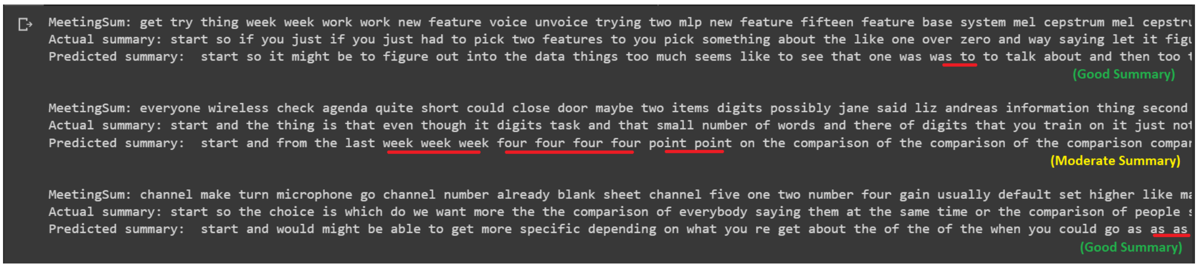


```
MeetingSum: almost forgot meeting twenty minutes ago thinking great brain something right news plans two weeks today less sicily couple three days flyi
Actual summary: start do not think we re probably from getting the system to understand things if we can get it to understand one thing like our where
Predicted summary:  start so it always that that it it to to to to understand the the of the same same at same at at the same at at ones ones ones ones

MeetingSum: talking today news conference talk yesterday yesterday morning video conference know talking nobody told anything talk supposed try decide
Actual summary: start they were trying to do something different like taking using filter that only even if they tell us that the are different we re s
Predicted summary:  start so it always that that it it to to to to to the the and and and and and and and he talk and and and and and and and the the r

MeetingSum: less get date johno meeting eva bhaskara add later like said adding thought write element situation nodes bayes net situation like mentione
Actual summary: start so like if just interested in the going there would just that information out of the that gets that would that would so we get at
Predicted summary:  start so if you be to to to to the the the of the of the same that that that we re are we re to the and and and and the the next we
```

Figure 5: Experiment 3

6.5 Experiment 4

The experiment 4 was performed using 70:30 training and testing dataset. It used latent dim as 300 and embedded dim as 200. The number of epochs is 50 and the batch size is 32. Maximum text length and maximum summaries kept as 470 and 220 respectively. The output as shown in Figure 6 showed moderate and good summaries. It was having fever repeated words and was more readable than the previous experiments, shown with red lines.



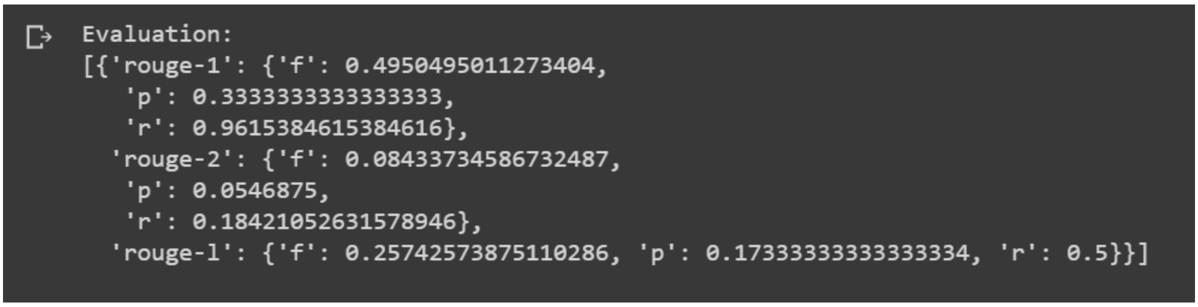
```
MeetingSum: get try thing week week work work new feature voice unvoice trying two mlp new feature fifteen feature base system mel cepstrum mel cepstru
Actual summary: start so if you just if you just had to pick two features to you pick something about the like one over zero and way saying let it figu
Predicted summary:  start so it might be to figure out into the data things too much seems like to see that one was was to to talk about and then too t

MeetingSum: everyone wireless check agenda quite short could close door maybe two items digits possibly jane said liz andreas information thing second
Actual summary: start and the thing is that even though it digits task and that small number of words and there of digits that you train on it just not
Predicted summary:  start and from the last week week week four four four four point point on the comparison of the comparison of the comparison compar

MeetingSum: channel make turn microphone go channel number already blank sheet channel five one two number four gain usually default set higher like ma
Actual summary: start so the choice is which do we want more the the comparison of everybody saying them at the same time or the comparison of people s
Predicted summary:  start and would might be able to get more specific depending on what you re get about the of the of the when you could go as as as
```

Figure 6: Experiment 4

6.6 Evaluation and Result

The evaluation was done using the qualitative (Human) and quantitative analysis (ROUGE) while analyzing the scores or metrics of the summary. There are various metrics to evaluate based on content based, co-selection based, text quality based etc. (Steinberger and Jezek; 2009). ROUGE score for the text summarization is used to evaluate the reference summary with the generated summary. ROUGE scores are of different types like ROUGE N (ROUGE1, ROUGE2), ROUGE L, ROUGE S and ROUGE W. It states how much reference summary and actual summaries have similarity between them.

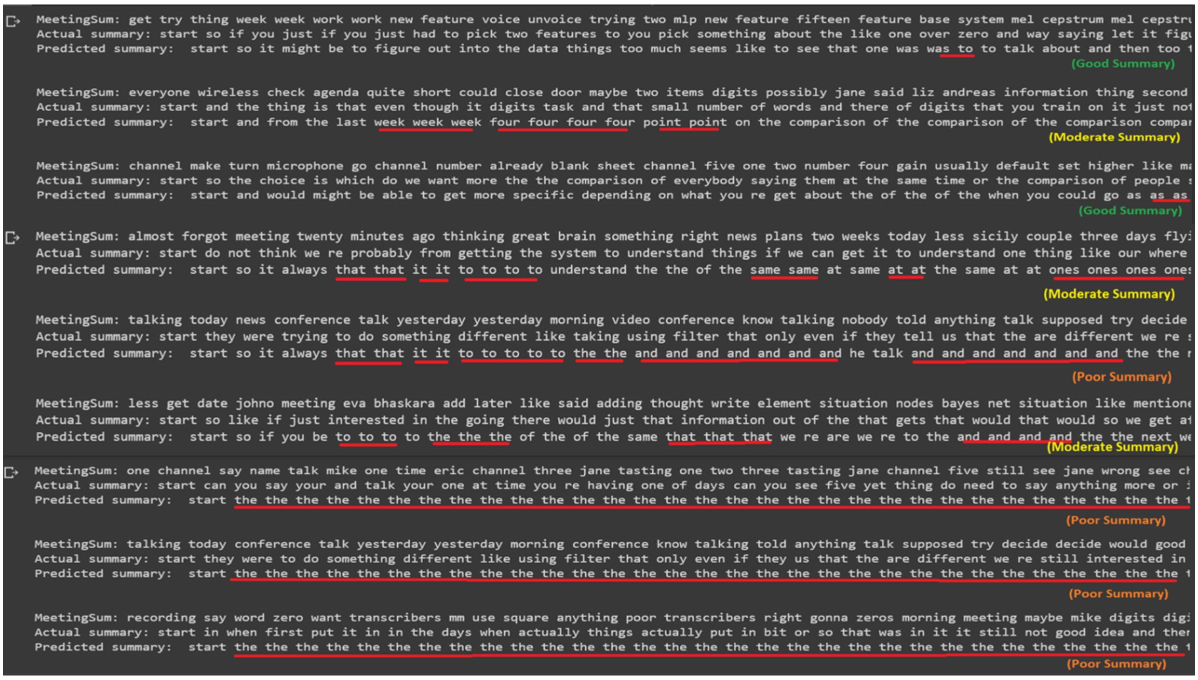
There was also human evaluation done in which 5 people evaluated the summaries randomly. As it was little difficult to read the long input text to know if the summary is



```
Evaluation:
[{'rouge-1': {'f': 0.4950495011273404,
   'p': 0.3333333333333333,
   'r': 0.9615384615384616},
  'rouge-2': {'f': 0.08433734586732487,
   'p': 0.0546875,
   'r': 0.18421052631578946},
  'rouge-l': {'f': 0.25742573875110286, 'p': 0.17333333333333334, 'r': 0.5}}]
```

Figure 7: Result of Final Experiment

generated correct or not. So, they reviewed reference summaries (extractive summary), actual summaries (abstractive summary), according to the ROUGE scores and mainly according to the human readability and understandability the output summaries was evaluated. The Figure 8 shows some of the evaluation.



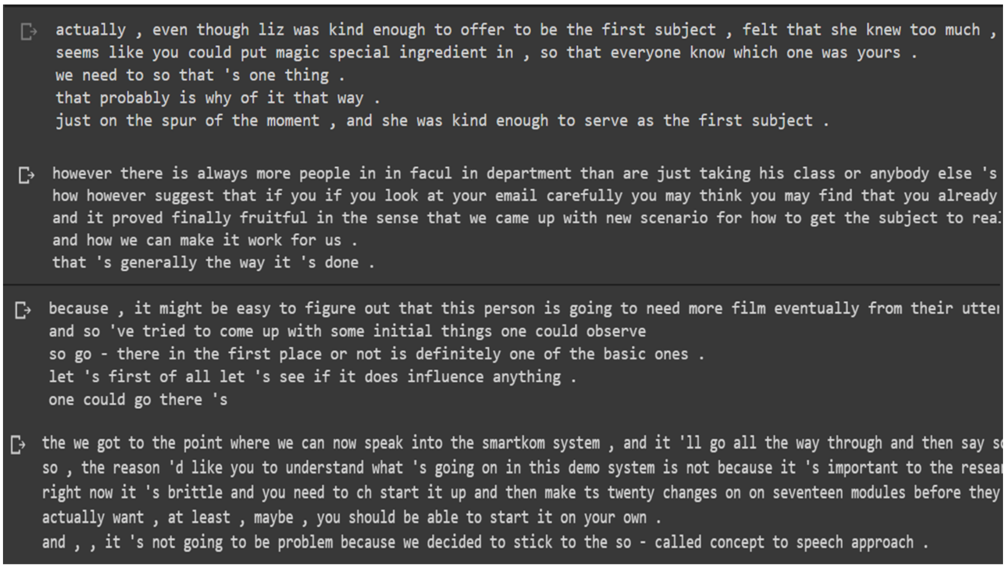
```
MeetingSum: get try thing week week work work new feature voice unvoice trying two mlp new feature fifteen feature base system mel cepstrum mel cepstru
Actual summary: start so if you just if you just had to pick two features to you pick something about the like one over zero and way saying let it figu
Predicted summary:  start so it might be to figure out into the data things too much seems like to see that one was was to to talk about and then too t
                                                                                                                                        (Good Summary)

MeetingSum: everyone wireless check agenda quite short could close door maybe two items digits possibly jane said liz andreas information thing second
Actual summary: start and the thing is that even though it digits task and that small number of words and there of digits that you train on it just not
Predicted summary:  start and from the last week week week four four four four point point on the comparison of the comparison of the comparison compar
                                                                                                                                    (Moderate Summary)

MeetingSum: channel make turn microphone go channel number already blank sheet channel five one two number four gain usually default set higher like ma
Actual summary: start so the choice is which do we want more the the comparison of everybody saying them at the same time or the comparison of people s
Predicted summary:  start and would might be able to get more specific depending on what you re get about the of the of the when you could go as as as
                                                                                                                                        (Good Summary)

MeetingSum: almost forgot meeting twenty minutes ago thinking great brain something right news plans two weeks today less sicily couple three days flyi
Actual summary: start do not think we re probably from getting the system to understand things if we can get it to understand one thing like our where
Predicted summary:  start so it always that that it it to to to to understand the the of the same same at same at at the same at at ones ones ones ones
                                                                                                                                   (Moderate Summary)

MeetingSum: talking today news conference talk yesterday yesterday morning video conference know talking nobody told anything talk supposed try decide
Actual summary: start they were trying to do something different like taking using filter that only even if they tell us that the are different we re s
Predicted summary:  start so it always that that it it to to to to to the the and and and and and and and he talk and and and and and and and the the r
                                                                                                                                        (Poor Summary)

MeetingSum: less get date johno meeting eva bhaskara add later like said adding thought write element situation nodes bayes net situation like mentione
Actual summary: start so like if just interested in the going there would just that information out of the that gets that would that would so we get at
Predicted summary:  start so if you be to to to to the the the of the of the same that that that we re are we re to the and and and and the the next we
                                                                                                                                    (Moderate Summary)
MeetingSum: one channel say name talk mike one time eric channel three jane tasting one two three tasting jane channel five still see jane wrong see ch
Actual summary: start can you say your and talk your one at time you re having one of days can you see five yet thing do need to say anything more or i
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                                        (Poor Summary)

MeetingSum: talking today conference talk yesterday yesterday morning conference know talking told anything talk supposed try decide decide would good
Actual summary: start they were to do something different like using filter that only even if they us that the are different we re still interested in
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                                        (Poor Summary)

MeetingSum: recording say word zero want transcribers mm use square anything poor transcribers right gonna zeros morning meeting maybe mike digits digi
Actual summary: start in when first put it in in the days when actually things actually put in bit or so that was in it it still not good idea and ther
Predicted summary:  start the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the the t
                                                                                                                                        (Poor Summary)
```

Figure 8: Evaluation done by Humans

Some of the results of extractive summaries output generated from the extractive approach are also shown in the Figure 9

6.7 Discussion

There are many important learnings and improvements to carry out from this research. For the summaries problems, it's not possible to predict results without a significant number of epochs and a large amount of data. Several small datasets have shown that the similar design can provide proper summaries since it's a much easier issue where one-to-one token mapping from a language to another is feasible. It is difficult to do so in an abstractive summary model. The model has learned to give more priority or higher probability score to START and END words just because they appear practically



```
actually , even though liz was kind enough to offer to be the first subject , felt that she knew too much ,
seems like you could put magic special ingredient in , so that everyone know which one was yours .
we need to so that 's one thing .
that probably is why of it that way .
just on the spur of the moment , and she was kind enough to serve as the first subject .

however there is always more people in in facul in department than are just taking his class or anybody else 's
how however suggest that if you if you look at your email carefully you may think you may find that you already
and it proved finally fruitful in the sense that we came up with new scenario for how to get the subject to rea
and how we can make it work for us .
that 's generally the way it 's done .

because , it might be easy to figure out that this person is going to need more film eventually from their utte
and so 've tried to come up with some initial things one could observe
so go - there in the first place or not is definitely one of the basic ones .
let 's first of all let 's see if it does influence anything .
one could go there 's

the we got to the point where we can now speak into the smartkom system , and it 'll go all the way through and then say s
so , the reason 'd like you to understand what 's going on in this demo system is not because it 's important to the resea
right now it 's brittle and you need to ch start it up and then make ts twenty changes on on seventeen modules before they
actually want , at least , maybe , you should be able to start it on your own .
and , , it 's not going to be problem because we decided to stick to the so - called concept to speech approach .
```

Figure 9: Sample of the generated summaries from extractive approach

in every line. But that is not to imply that the model isn't good; it just needs to have more epochs and need training on more input data. Using CUDNN LSTM throughout this implementation of the LSTM gives more sense because it is quicker in calculations. Lengthier sentences should be avoided for the issue since LSTM are vulnerable to vanish gradient issues. If the batch size is known ahead of time, LSTM (stateful) can be used, which increases performance slightly. The output from LSTM, cell states and backward-forward hidden states are all returned by the bidirectional wrapper. To conserve memory, the useless parameters must not have given names. Learning rate, batch size, context vector, LSTM size and number of epochs show a significant influence on hyperparameters. Extractive text summarization process based on the frequency are easy and simple to perform. As they rely on the frequency of occurrence of every word and it requires a larger database to determine the right summary. ROUGE metrics are appropriate for extractive summarizing but not for the abstractive summarizing.

7 Conclusion and Future Work

In this paper, the challenge of abstractive text summarization for constructing a brief and non-repetitive summary was examined with the extractive approach, which is a rarely studied topic in the existing research because it's difficult to obtain good summaries through abstractive. The majority of previous research focuses on overcoming the challenge of producing a summary that is shorter than the original text but ignores the importance of producing accurate, non-repetitive and useful summaries. According to the literature review, it states that mostly all worked on small input having very fewer lines and generated the readable or short output summary from a single document.

In this research, a novel use of the hybrid approach for text summarization that includes both the abstractive and extractive approach to work with huge data. It combined merits of both the summarization using various techniques such as TextRank, RNN, and LSTM with attention mechanism which I found appropriate according to the meeting domain, type and number of transcripts. I developed the model on the ICSI corpus datasets which contains numbers of transcripts with long conversation between participants that made this more difficult to implement or generate summaries because of lengthy input transcript and limited computing size. The result obtained from the extractive summarization in the first phase was better and after giving to the abstractive model in the second phase, it generated more repetitive words for a few of the summaries shown in section 6 and were not readable. Numerous experiments were carried out which later on reduced the repetitive words and made it readable, but still some of the summaries were not generated properly, having higher number of repetition in them.

For future work, the research will be more focused on other abstractive techniques and minimizing the repetitive words to negligible in a summary and dealing with computational error that occurred because of huge data. Here, using rather than 1 hot encoding of the decoder's softmax outputs, categorical cross entropy might possibly save a large amount of memory. For the similar model, when trained on the machine translation issue on a small dataset and with one epoch, it generates the repeated words and increases the performance significantly. As a result, it's reasonable to assume that, with more data and time, the model might provide excellent summaries. Still, implementation is difficult, and obtaining the optimal results requires several experiments. For summarizing jobs, human-based assessment is still required. Further, improvements are discussed in the discussion section 6.7. By evaluating several Recurrent Neural Network (RNN) implementations of extractive text summarization, also be done to develop a pipeline for text data pre-processing. The abstractive text summarization models will be improved later on in order to achieve better results. For extractive and abstractive summarization, evaluation based on embedding based techniques to be carried out in the future.

References

Allen, J. A., Sands, S. J., Mueller, S. L., Frear, K. A., Mudd, M. and Rogelberg, S. G. (2012). Employees' feelings about more meetings: An overt analysis and recommendations for improving meetings, *Management Research Review* .

Andhale, N. and Bewoor, L. (2016). An overview of text summarization techniques, *2016 International Conference on Computing Communication Control and automation (ICCUBEA)*, pp. 1–7.

Badgujar, C., Jethani, V. and Ghorpade, T. (2018). Abstractive summarization using graph based methods, *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, pp. 803–807.

Balasundaram, K. and Amalraj, C. (2019). Speech document summarization using neural network, *2019 4th International Conference on Information Technology Research (ICITR)*, pp. 1–3.

Bokaetf, M. H., Sameti, H. and Liu, Y. (2015). Unsupervised approach to extract summary keywords in meeting domain, *2015 23rd European Signal Processing Conference (EUSIPCO)*, pp. 1406–1410.

Buist, A., Kraaij, W. and Raaijmakers, S. (2004). Automatic summarization of meeting data: A feasibility study.

Gunawan, D., Harahap, S. H. and Fadillah Rahmat, R. (2019). Multi-document summarization by using textrank and maximal marginal relevance for text in bahasa indonesia, *2019 International Conference on ICT for Smart Society (ICISS)*, Vol. 7, pp. 1–5.

Hahn, U. and Mani, I. (2000). The challenges of automatic summarization, *Computer* **33**: 29–36.

Haider, M. M., Hossin, M. A., Mahi, H. R. and Arif, H. (2020). Automatic text summarization using gensim word2vec and k-means clustering algorithm, *2020 IEEE Region 10 Symposium (TENSYMP)*, pp. 283–286.

Hao, Z., Ji, J., Xie, T. and Xue, B. (2020). Abstractive summarization model with a feature-enhanced seq2seq structure, *2020 5th Asia-Pacific Conference on Intelligent Robot Systems (ACIRS)*, pp. 163–167.

JUGRAN, S., KUMAR, A., TYAGI, B. S. and ANAND, V. (2021). Extractive automatic text summarization using spacy in python amp; nlp, *2021 International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE)*, pp. 582–585.

Kaur, M. and Srivastava, D. (2019). Text summarization using partial textual entailment based graphs, *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, pp. 366–374.

Liu, F., Liu, F. and Liu, Y. (2011). A supervised framework for keyword extraction from meeting transcripts, *IEEE Transactions on Audio, Speech, and Language Processing* **19**(3): 538–548.

Madhuri, J. and Ganesh Kumar, R. (2019). Extractive text summarization using sentence ranking, *2019 International Conference on Data Science and Communication (IconDSC)*, pp. 1–3.

Merchant, K. and Pande, Y. (2018). Nlp based latent semantic analysis for legal text summarization, *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1803–1807.

Mohammad Masum, A. K., Abujar, S., Islam Talukder, M. A., Azad Rabby, A. S. and Hossain, S. A. (2019). Abstractive method of text summarization with sequence to sequence rnns, *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pp. 1–5.

Padmanandam, K., Bheri, S. P. V. D. S., Vegesna, L. and Sruthi, K. (2021). A speech recognized dynamic word cloud visualization for text summarization, *2021 6th International Conference on Inventive Computation Technologies (ICICT)*, pp. 609–613.

Rahul, Adhikari, S. and Monika (2020). Nlp based machine learning approaches for text summarization, *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 535–538.

Romano, N. and Nunamaker, J. (2001). Meeting analysis: Findings from research and practice., p. 13 pp.

Shirwandkar, N. S. and Kulkarni, S. S. (2018). Extractive text summarization using deep learning, *2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)* pp. 1–5.

Steinberger, J. and Jezek, K. (2009). Evaluation measures for text summarization., *Computing and Informatics* **28**: 251–275.

Widyassari, A. P., Rustad, S., Shidik, G. F., Noersasongko, E., Syukur, A., Affandy, A. and Setiadi, D. R. I. M. (2020). Review of automatic text summarization techniques methods, *Journal of King Saud University - Computer and Information Sciences* .
URL: *https://www.sciencedirect.com/science/article/pii/S1319157820303712*

Xie, S. and Liu, Y. (2011). Using n-best lists and confusion networks for meeting summarization, Vol. 19, pp. 1160–1169.

Zhang, Justin Jian, F. P. (2012). Automatic parliamentary meeting minute generation using rhetorical structure modeling, *IEEE Transactions on Audio, Speech, and Language Processing* **20**(9): 2492–2504.