**` Department of Computer Science & Engineering**

*A Project Report on*

**Multimedia Text Summary Generator**

**for the Visually Impaired**

*Submitted in partial fulfillment of the requirements for the award of the degree of*

**Bachelor of Engineering in Computer Science & Engineering**

*By*

DEEPTHI S SHETTY 1MS17CS027

K V SANDEEP 1MS17CS045

MANOJ D 1MS17CS059

P DHANUSHA 1MS17CS078

*Under the guidance of*

Dr. Jayalakshmi D. S.

Associate Professor

**M S RAMAIAH INSTITUTE OF TECHNOLOGY**

**(Autonomous Institute, Affiliated to VTU)**

**BANGALORE-560054**

[**www.msrit.edu**](http://www.msrit.edu/)

July 2021

**Department of Computer**

**Science & Engineering**

**CERTIFICATE**

Certified that the project work entitled “Multimedia Text Summary Generator for the Visually Impaired” carried out by DEEPTHI S SHETTY – 1MS17CS027, K V SANDEEP - 1MS17CS045, MANOJ D - 1MS17CS059, P DHANUSHA – 1MS17CS078 are bonafide student of M.S.Ramaiah Institute of Technology Bengaluru in partial fulfillment for the award of Bachelor of Engineering in Computer Science and Engineering of the Visvesvaraya Technological University, Belgavi during the year 2020-21. It is certified that all corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library.

The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

**Project Guide** **Head of the Department**

**DR. JAYALAKSHMI D. S. DR. ANITA KANAVALLI**

**External Examiners**

**Name of the Examiners: Signature with Date**

**1.**

**2.**

**Department of Computer**

**Science & Engineering**

**DECLARATION**

We, hereby, declare that the entire work embodied in this project report has been carried out by us at M. S. Ramaiah Institute of Technology, Bengaluru, under the supervision of **Dr. JAYALAKSHMI D. S, Associate Professor,** Dept of CSE. This report has not been submitted in part or full for the award of any diploma or degree of this or to any other university.

Signature Signature

DEEPTHI S SHETTY K V SANDEEP

1MS17CS027 1MS17CS045

Signature Signature

MANOJ D P DHANUSHA

1MS17CS059 1MS17CS078

**ACKNOWLEDGEMENT**

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project. We would like to express our profound gratitude to the Management and **Dr. N.V.R Naidu** Principal, M.S.R.I.T, Bengaluru for providing us with the opportunity to explore our potential.

We extend our heartfelt gratitude to our beloved **Dr. Anita Kanavalli**, HOD, Computer Science and Engineering, for constant support and guidance.

We whole-heartedly thank our project guide **Dr. Jayalakshmi D.S,** for providing us with the confidence and strength to overcome every obstacle at each step of the project and inspiring us to the best of our potential. We also thank him/her for his/her constant guidance, direction and insight during the project.

This work would not have been possible without the guidance and help of several individuals who in one way or another contributed their valuable assistance in preparation and completion of this study.

Finally, we would like to express sincere gratitude to all the teaching and non-teaching faculty of CSE Department, our beloved parents, seniors and my dear friends for their constant support during the course of work.

DEEPTHI S SHETTY K V SANDEEP

MANOJ D P DHANUSHA

**ABSTRACT**

With the advancing methodologies in the field of NLP, text summarization has become an important application and is still under research. Taking into consideration today’s busy world, time is a most important factor for everyone, and people don’t bother to make time to listen to long audio news or read long news articles and as the visually impaired are an important part of our society, it becomes still difficult for them to read although there is braille but it is inconvenient for them every time to read through braille. The aim is to make use of advancing technology to make their lives easier.

Our project also aims to summarize multiple documents of same source content and summarize it so as to get a brief insight of the information from various sources of the same data. With the amount of data being generated each day increasing exponentially it is difficult for us to know about everything so in order to get a summary of the data this project helps us do it. Here we take multiple sources of same data having same opinion as the input and summarize it and the output will be available in both audio and text format. The summarization here is implemented using various NLP algorithms like BERT, BART, GP T2.This project also helps to analyze the accuracy scores of the algorithms in summarizing the text and help us in improvising the quality of the summary generated.

This gave us inspiration that led to the idea of generating concise and short summaries for the visually impaired. Our system makes use of various APIs like speech-recognition, pyspeech, Google Cloud Speech API etc. to extract text and then use summarization techniques to present the most accurate summary and convert it back to audio format so that the news is more accessible for the visually impaired.

**TABLE OF CONTENTS**

**Chapter No. Title Page No.**

Acknowledgement…………………………………………………………………iv

Abstract……………………………………………………………………………v

1. INTRODUCTION

* 1. General Introduction………………………………………………………..3
  2. Problem Statement………………………………………………………….4
  3. Objectives of the Project……………………………………………............4
  4. Project Deliverables………………………………………………………...5
  5. Current Scope……………………………………………………………….5
  6. Future Scope………………………………………………………………...5

2. PROJECT ORGANISATION

2.1 Software Process Model……………………………………………………6

2.2 Roles and Responsibilities………………………………………………….6

3. LITERATURE SURVEY

3.1 Introduction………………………………………………………………...7

3.2 Related works with the citation of the References…………………………7

3.3 Conclusion of the Survey…………………………………………………..13

4. PROJECT MANAGEMENT PLAN

4.1 Schedule of the Project……………………………………………………14

4.2 Risk Identification………………………………………………………...14

5. SOFTWARE REQUIREMENT SPECIFICATIONS

5.1 Product Overview…………………………………………………………15

5.2 External Interface Requirements

5.2.1 User Interfaces……………………………………………………….15

5.2.1 Hardware Interfaces………………………………………………….15

5.2.2 Software Interfaces…………………………………………………..15

5.2.3 Communication Interfaces…………………………………………...16

5.3 System Features

5.3.1 Functional Requirements……………………………………………16

5.3.2 Non-Functional Requirements……………………………………....17

6. DESIGN

6.1 Introduction……………………………………………………………….18

6.2 Architecture Design……………………………………………………....18

6.3 Graphical User Interface………………………………………………….19

6.4 Class Diagram and Classes……………………………………………….20

6.5 Sequence Diagram ……………………………………………………….20

6.6 Data flow diagram………………………………………………………..21

* 1. Conclusion………………………………………………………………..21

7. IMPLEMENTATION

7.1 Technology Introduction…………………………………………………22

7.2 Overall view of the project in terms of implementation…………………22

7.3 Explanation of Algorithm and how it is been implemented……………..23

7.4 Information about the implementation of Modules……………………...25

8. TESTING

8.1 Introduction………………………………………………………….…..27

8.2 Test cases…………………………………………………………….….27

1. RESULTS & PERFORMANCE ANALYSIS

9.1 Result Snapshots…………………………………………………………31

9.2 COMPARISION results tables ………………………………………….34

9.3 Performance analysis – graphs, tables etc……………………………….35

1. CONCLUSION & SCOPE FOR FUTURE WORK

10.1 Significance of the Proposed Research Work………………………….42

10.2 Directions for the Future works ………………………………………..42

**REFERENCES**

**APPENDICES:**

1. Software manual (if required in the project)
2. Data Set
3. Publications
4. Certificates

**LIST OF FIGURES**

Figure 1. Gantt chart……………………………………………………………….14

Figure 2.1 Architecture Design……….…….……………………………………..18

Figure 2.2 Class Diagram………………………………………………………….20

Figure 2.3 Sequence Diagram……………………………………………………..20

Figure 6.4 Data flow Diagram …………………………………………………….21

Fig 9.1.1 New input article for GPT-2 (single document summarizer) ………...…31

Fig 9.1.2 Expected Summary (Human-generated) ………………………………..31

Fig 9.1.3 GPT-2 Summary (Computer-generated) ……………………………..…32

Fig 9.1.4 Multi-Document summarization: Document-1…………………………..32

Fig 9.1.5 Multi-Document summarization: Document-2…………………………..32

Fig 9.1.6 Multi-Document summarization: Document-3…………………………..33

Fig 9.1.7 Multi-Document summarization: Human written summary…………….33

Fig 9.1.8 Multi-Document summarization: Computer generated summary…….…33

Fig 9.1.9 Audio as input news and generated audio summary………………….…33

Fig 9.3.1 BART for single document summarization……………………………..35

Fig 9.3.2 BERT for single document summarization……………………………...35

Fig 9.3.3 GPT-2 for single document summarization……………………….….…36

Fig 9.3.4 Comparison of BERT, BERT, GPT-2 for single document

Summarization………………………………………………………………..……36

Fig 9.3.5 Rouge scores for Multiple document summarization using BART

with Cosine Similarity……………………………………………………..………37

Fig 9.3.6 Rouge scores for Multiple document summarization using BERT

With Cosine Similarity…………………………………………………………….37

Fig 9.3.7 Rouge scores for Multiple document summarization using GPT-2 with

Cosine Similarity…………………………………………………………………..38

Fig 9.3.8 Rouge-1 scores for comparison of BERT, BART and GPT-2 in multi

document summarization (Cosine similarity)…………………………….………..38

Fig 9.3.9 Rouge scores for Multiple document summarization using BART with

Jaccard Similarity………………………………………………………………….39

Fig 9.3.10 Rouge scores for Multiple document summarization using BERT with

Jaccard Similarity………………………………………………………………….39

Fig 9.3.11 Rouge scores for Multiple document summarization using GPT-2 with

Jaccard Similarity…………………………………………………………………..40

Fig 9.3.12 Rouge-1 scores for comparison of BERT, BART and GPT-2 in multi

document summarization (Jaccard similarity)………………………………………40

Fig 9.3.13 Rouge scores for mixed multi document summarization using GPT-2…41

Fig 9.3.14 Rouge-1 scores for comparison of GPT-2 with cosine similarity, GPT-2

with Jaccard similarity and GPT-2 with mixed summarization…………………….41

**LIST OF TABLES**

Table 1. Roles and Responsibilities……………………………………………….6

Table 2. Software Interfaces………………………………………………………15

Table 9.2.1 Single Document Summarization…………………………………….34

Table 9.2.2 Multi-Document Summarization……………………………………..34

Table 9.2.3 News with different opinion vs normal news…………………………34

Table 9.2.4 Comparison for 3 articles and input articles…………………………..34

**Chapter 1**

**INTRODUCTION**

* 1. **General Introduction**

Text summarization has gained a lot of importance in recent times because of the excess of information available online and the lack of time in people’s lives to give enough attention to it. This is resulting in increasing demand of more capable text summarizers. As the volume of data being generated and consumed is increasing day by day, representing information in the smallest possible form while preserving its meaning has become an important task.

With many different sources available for same data, it therefore becomes a tough decision to make a choice about source. This is what leads into multi document summarization, which summarizes a topic for you collecting data from different sources available. It is a tedious task to read long articles to get all the relevant information, instead, reading well-formed summaries to get the maximum information is way more convenient.

Sometimes many people feel it difficult to listen to some long talks and would instead like to get a gist of the talk, which introduces audio summarization that summarizes a long audio file so as to listen only to the main points of talk. Sadly, the huge number of visually impaired and outwardly debilitate

d individuals can't straightforwardly or helpfully read normal news reports like located individuals, and they need to peruse braille with their fingerprints but it's an extremely inefficient methodology because it is absolutely time taking and needs several steps to be followed.

Our goal is to summarize speeches or articles in audio or text format by applying various NLP techniques and convert them into audio format so that the information is conveyed effectively to the visually impaired.

**1.2 Problem Statement**

It is often inconvenient for the blind to read huge articles and speeches or even for a normal person to listen long audios or read huge articles. The lack of those individuals to browse text incorporates an immense impact on their lives. They do have some techniques available like Braille but its highly inconvenient for them and is absolutely time taking and needs several steps to be followed.

We are aiming to build “Multimedia Text Summary Generator for Visually Impaired”.

The main aim of the Multimedia Text Summary Generator is to get summarized outputs for the input texts/audios. We use NLP models here to summarize the text and then convert it back to audio with the help of API’s. The audio is preferred as an output as it is generally easier for impaired to get insights of the information. This can also be used by people who are busy and are interested in knowing about things in short. It helps them save time and also get brief insights about the information.

Our system could take input in both text and audio format. The system is easily accessible by both normal and visually impaired. The output summary is in the audio format. This makes it convenient for the visually impaired to get information quickly. Multi-document summarization helps people avoid reading multiple newspapers to acquire the information as it summarizes on a topic collecting data from different sources. It can also be made use to summaries the movie reviews.

* 1. **Objectives of The Project**

The objective is to save time and effort by automating the process of generating accurate and short summaries.

Our goal is to summarize speeches and convert them into audio format so that the information is conveyed effectively to the visually impaired. The input to our model can be either the speeches or articles in text or audio format, and with this input in order to present the summary we use models like BERT, BART, GPT-2 to summarize the text to a precise and short summary and converting it back to audio format to make it easily accessible to the visually impaired. The audio form of input is mainly focused on so that it is convenient for the visually impaired to get information quickly. They can save time and listen to the short version any time while travelling or doing some other work.

**1.4 Project Deliverables**

Our project aims to create a summarizer for visually impaired people which helps them to understand the summary of the text in audio format. It also summarizes the audio inputs and gives the summary of it.

**1.5 Current Scope**

* The visually impaired are an integral part of our society. The thought was to make use of the technological advances to make their lives easier.
* This led to the idea of generating concise summaries for the visually impaired.
* We are trying to implement NLP summarization techniques on the text documents and text extracted from audio files.
* The aim is to convert audio/text files into an audio summary.
* Additionally, we are also planning to summarize multiple documents to a single short summary.
* To make it user friendly we are converting the generated summary back to audio format which will be easier to listen to anywhere and anytime.

**1.6 Future Scope**

Future work might include:

* Bigger dataset can be built for multi-document summarization.
* Better audio summarization.

**Chapter 2**

**PROJECT ORGANISATION**

**2.1 Software Process Model**

Process model used in building the software is the agile model - Scrum model.

Agile model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software products. Agile Methods break the product into small incremental builds. These builds are provided in iterations, Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements.

The Scrum model suggests that projects progress via a series of sprints. Scrum methodology advocates for a planning meeting at the start of the sprint, where team members figure out how many items they can commit to, and then create a sprint backlog a list of the tasks to perform during the sprint. On each day of the sprint, all team members should attend a daily Scrum meeting of at least 15 minutes.

**2.2 Roles and Responsibilities**

Table 1. Roles and Responsibilities

|  |  |
| --- | --- |
| **Roles** | **Responsibilities** |
| K V Sandeep | Research of data to be used and NLP methods,  Fine tuning of algorithms |
| Deepthi S Shetty | Validating the dataset available and removing all redundancies,  Implementation of 3 algorithms |
| Manoj D | Research of API's to be used and algorithms,  Fine tuning of algorithms |
| P Dhanusha | Implementation of API's being used,  Implementation of 3 algorithms |

**Chapter 3**

**LITERATURE SURVEY**

**3.1 Introduction**

A huge number of visually impaired individuals can’t read news reports like sighted individuals, and they need to peruse braille with their fingerprints but it’s an extremely inefficient methodology because it takes a lot of time and needs several steps to be followed. Our goal is to summarize speeches or articles in audio or text format by applying various NLP techniques and convert them into audio format so that the information is conveyed effectively to the visually impaired.

**3.2 Related Works with the citation of the References**

**3.2.1 Text summarization using transfer learning**

This section deals with the automate text summarization by using two approaches: extractive and abstractive.

A summary of a long text document enables people to easily grasp the information of the topic without having the need to read the whole document. This thesis aims to automate text summarization by using two approaches: extractive and abstractive. The former approach utilizes submodular functions and the language representation model BERT, while the latter uses the language model GPT-2.

The first step in creating an extractive summary is to split the input text into sentences and finding vector representations of each sentence. BERT is used to extract the high dimensional context vector for every sentence. By doing this, every sentence in the entire text document becomes their own representation where similar context vectors represent similar sentences.GPT-2 is used for abstractive summarization. This model layout is never altered and is used as is, fine-tuned to perform abstractive summarization.

The results obtained using the GPT-2 on the dataset are competitive to state-of-the-art. Besides the quantitative evaluation, a qualitative investigation in the form of a human evaluation was performed, along with inspection of the trained model that demonstrates that it learns reasonable abstractions.

**3.2.2 Critical Evaluation of Neural Text Summarization**

This section deals with critical evaluation of the current research setup for text summarization which mainly includes the datasets, the evaluation metrics and the models.

In most of the summarization experiments conducted till now, the majority of datasets deal with the news domain. Some of the most common news datasets used are CNN/Daily Mail, XSum. Outside the news domain, TIFU (collection of posts scraped from reddit, TL; DR summary) is one of the most popular datasets. The problem with the current datasets used is:

1. Summarization is mostly done as an under constrained task. Assessing important information depends on prior knowledge of the user. A study was conducted to demonstrate the difficulty and ambiguity of content selection. A constrained summary is more precise whereas an under constrained summary is more verbose.

2. Data scraped from the web might contain noise. The quality is highly dependent on the preprocessing steps. Manual inspection to check the noise is tough.

The most popularly used evaluation tool for summaries is the ROGUE package. The basis of automatic metrics offered by this package is the grammatical overlap between candidate and reference summaries and is based on exact token matches. But, the issue is that the overlap between phrases that are grammatically different but actually mean the same is not supported. Many extensions of the ROGUE package are available such as ParaEval, ROGUE-WE, ROGUE 2.0 and ROGUE-G.

The datasets require additional constraints for the creation of good summaries. Layout bias is one of the major factors that affect the result of current methods in a negative way. The current evaluation protocol is weakly correlated to human judgments and fails to evaluate important features such as factual correctness of the summary.

## **Pre trained Language Models**

The section discusses detailed implementation of a pretrained language model to the application of text summarization.

Pretrained language models can be fine-tuned to do various task-specific jobs. The aim is to use a pre-trained language model to compress a document into a shorter form while retaining most of the important information. The section deals with three main things: document encoding, ways to effectively employ pretrained models and proposed models that can be used for the task of summarization.

Pretrained language models expand the concept of words embeddings by learning contextual representations. BERT is a advanced representational language model which is trained along with masked language modeling. The general architecture can be described as follows: Input text is given as input through three embeddings, First one being Token embeddings, which indicates the meaning of every token. Second one is Segmentation embeddings which discriminates between two sentences. The last one is Position embeddings that indicates the position of each token within the text sequence. Single input vector is formed by adding these embeddings and is fed to a bidirectional transformer. The final output is the output vector for each token with contextual information.

ROGUE packages R-1 and R-2 are used to assess instructiveness whereas RL(LCS) is used to assess fluency. BERT based models outperform other models across all datasets.

## **BART-based Approach for Document Summarization**

Nowadays, researchers have been increasingly tasked by funders and publishers to outline their research for the public by writing a lay summary. Therefore, it is essential to automatically generate lay summaries to reduce the workload for researchers as well as build a bridge between the public and science.

In this paper, a lay summary generation system was built based on the BART model. leverage sentence labels were used as extra supervision signals to improve the performance of lay summarization.

BART is based on the standard Transformer model which can be regarded as generalizing BERT (due to the bidirectional encoder), GPT (with the left-to-right decoder). It is pre-trained on the same corpus as RoBERTa with two tasks: text infilling and sentence permutation.

BART obtains great performance on the summarization task. The BART fine-tuned was used on CNN/DailyMail dataset in this model

**3.2.5 Two-level Text Summarization from Online News Sources with Sentiment Analysis** The flow of the methodology of the project goes by Pre-Processing of data and then Generate First-Level Summary and then perform Sentimental Analysis and Generate Second-Level Summary.

This research uses ROUGE-1 and ROUGE-2 methods, which are n-grams to evaluate the First-level summaries. The Recall, Precision and F-Measure is calculated by using less sentences (LS) and more sentences (MS).It shows more than 60% Recall in Movie Review from ROUGE-1 even with more than 90% of reduction in their summary. The two-level summary provides only the important content from various related online news articles on a news topic in one place.As a future work, combining this Extraction-based method with Abstraction-based method would enhance the news summarization results. Building this research in the form of an Android and IOS applications would greatly help many people who use cell phones to read the news.

* + 1. **Single Document Summarization using Sentence Embeddings and K-Means Clustering**

In the following Research work the clustering algorithm used sentence embeddings. One sentence from each cluster was selected as its representative. This selection was done by the ridge regression based sentence scoring model . The summary generated contained the selected representative sentences. The approach can be listed as the following three steps:(A)Calculating sentence embeddings(B)Clustering(C) Summary generation.

For ROUGE score calculation, we used Porter stemming to pre-process the generated and actual summaries. Both metrics were calculated on summaries of varying lengths, i.e. 1 sentence to 5 sentences long. The upper limit was set as 5 because DUC 2001 contains multiple documents with a total number of sentences equal to 5. It was found that the proposed approach produced satisfactory results.The future work can be to use self-trained sentence embeddings for context based text summarization and selection of the representative sentence from each cluster from a pre-trained sentence scoring model can be used for extractive single-document summarization.

* + 1. **Automatic Text Summarization and it’s methods - A Review**

The following paper was an analysis paper based on various methods.The implementation of the paper is done in stages as follows-

(A)Term Frequency Based Method

(B)Graph Based Method

(C)Time Based Method

(D)Clustering Based Method

(E)Semantic Dependency Based Method

(F)Topic Based Approaches

(G)Discourse Based Approaches

(H)LSA Based Approaches

(I)Approach Based on Fuzzy Logic

As a final point, particular contemporary inclinations in automatic appraisal of summarization schemes have been fathomed. The truncated inter-annotator preparation figures pragmatic throughout manual estimations put forward upcoming experiments of this problem area profoundly be contingent in capacity to find competent approach for inevitably weighing such kind of text summarization system, and also on enlargement of procedures which impartial adequate to ordinarily putative by the peoples who are doing researches on this area.

This detailed assessment accentuates extractive methodologies to summarization using numerous approaches. A discrepancy has occurred amongst single/solitary document and multi-document summarization. Subsequently, some motivating efforts are done up to now within earlier research, This work preferred towards embrace a transitory conversation for few good approaches that we found to be of good scope for future research, also, these works are emphasis only on trivial minutiae associated with a wide-ranging summarization progression and not on proposing an all-inclusive summarization scheme.

**3.2.8. Study on Abstractive Text Summarization Techniques**

The text summarization has seen two approaches, extractive text summarization, and abstractive text summarization. Both abstractive and extractive text summarization can be used for various applications. Therefore, in order to get corresponding data as per the application's purpose, easily and quickly from different sources of data on the internet, an online content summarizer is desired. Summarizers makes it easier for users to understand the content without reading it completely. Abstractive Text Summarizer helps in defining the content by considering the important words and helps in creating summaries that are in a human-readable format. The main aim is to make summaries in such a way that it should not lose its context.

The model proposed in the paper extends the state of-the-art model for abstractive sentence summarization to recurrent neural network architecture. This model is a simplified version of the encoder-decoder framework for machine translation. In this case, the encoder input is a sequence of words which are subsequently converted into a vector representation and the decoder, assisted by the attention mechanism which focuses on specific words at each step of the input sequence. The model is trained on the Gigaword corpus to generate headlines based on the first line of each news article. This model uses Conditional probability to generate a target sequence of tokens, which is calculated based on sentence summary pairs. Convolutional and attention-based recurrent neural network models are used for the problem of abstractive sentence summarization.

This paper has explained the basis of RNN models used for the development of attention models, as well as gives a brief insight into feature selection, attention model, pointer mechanism, and how in synergy they produce abstractive text summaries. The current model does not work if multiple documents are passed to the model. In the future, a mechanism can be devised to preserve the context of the previous document before taking into account the next document.

**3.2.9. An Automatic Multi-document Text Summarization Approach Based on Naive Bayesian Classifier Using Timestamp Strategy**

The paper deals with a set of seven multistage compression steps are introduced:

(1) Set of documents is provided for processing.

(2) From the set of documents, frequently used related documents are selected by the system for processing.

(3) Preprocessing work is done and the sentences are broken into words.

(4) Score is calculated for each word using Bayesian classifier.

(5) Score is calculated for each sentence.

(6) For each sentence group, one sentence level illustration is selected.

(7) The sentence level illustration is either generated as a linear sentence or further compressed if necessary.

It is observed that the summary generated by the system has all important contents as 90% of the human assessors stated that the summary generated by the system is comprehensible. From this analysis done using human assessors it is proved that the summary generated by the system is short and the quality of the summary generated is also good based on the two factors readability and comprehensibility.

The proposed work does not involve a knowledge base and therefore can be used to summarize articles from fields as diverse as politics, sports, current affairs, and finance. However, it does cause a tradeoff between domain independence and a knowledge based summary which would provide data in a form more easily understandable to the human user. A possible application of this work can be made to make data available on the move on a mobile network by even shortening the sentences produced by our algorithm and then shortening it.. This will ensure that the summary produced is to the highest condensed form which can be made in the mobile industry.

**3.2.10.Multi-document summarization using sentence clustering**

The paper explains the following process which includes the following steps-

1. Preprocessing

2. Feature extraction

3. Single document summary generation

4. Multi-document summary generation

The researches have conducted two experiments to evaluate their system. First experiment evaluates single document summarization. The second experiment focuses on the evaluation of multi document summarization. The evaluation was done using the DUC 2002 dataset.

They observed an average recall of 0.45947, an average precision of 0.47989 and F-measure of 0.46768 on the DUC 2002 dataset. The best result reported on the DUC 2002 dataset had a recall of 0.4804. Multi-document summarization using sentence clustering gives the recall of 0.33358, precision of 0.34221 and F-measure of 0.33774. The results obtained on multi-document summarization using sentence clustering are more promising.

This paper presents a method for multi document summarization by combining single document summaries. The features used for generating single document summaries are sentence weight, sentence reference index feature, location feature and concept similarity feature. The single document summaries using sentence clustering method to generate multi document summary.. A number of semantic similarity measures based on these concepts exist in literature, which can be used to decide semantic similarity between sentences. The syntactic similarity used in this paper is based on word order, which can be replaced with other structural similarity measures. We would also like to evaluate our system on DUC 2005 or DUC 2006 datasets for query-based summarization.

**3.3 Conclusion of Survey**

To summarize articles using various Natural language summarizing techniques which includes various methods like BERT, BART, GPT-2. Different methods of summarization were applied on the dataset and these methods were evaluated using ROUGE metric to decide upon the best method for summarizing the articles. To also present multi document summarization where similar topic articles from different sources are collected and applying

.

**Chapter 4**

**PROJECT MANAGEMENT PLAN**

**4.1 Schedule of the Project**

**GANTT CHART**

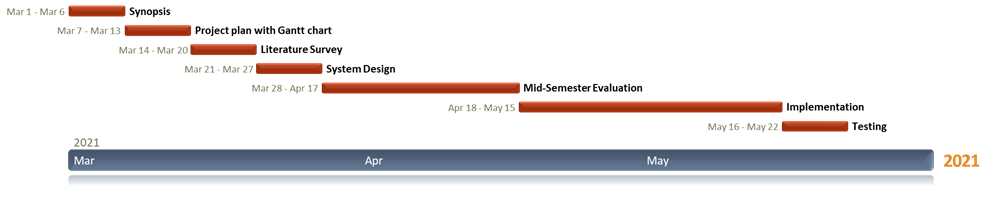


Figure 6. Gantt chart

**4.2 Risk Identification**

Speech recognition in Kannada is quite challenging due to the following reasons:

1. As this is an NLP based project, cannot guarantee which method will give the best results. Speech signal characteristics distorted due to varying room acoustics, channel characteristics, microphone characteristics, and background noise.
2. The data collected play’s crucial part, if the summaries collected are from same author, then the model might get biased. The challenge is to create short summaries while maintaining the content quality and fluency.

With the above challenges and risks involved, we try to consider every point and make sure our implementation will yield a better result.

**Chapter 5**

**SOFTWARE REQUIREMENT SPECIFICATIONS**

**5.1 Product Overview**

We start with a data corpus (data collection) with a good size of data. After preparing a data corpus, the corpus is then used to train the system as well as to test it. The data can be either audio files or text files if audio then needs to be converted to text and then processed. Search for same articles from different sources is not easy as these data is not available in large quantities. Once the model is trained it can convert a new news article into a small summary and read out to the visually impaired

**5.2 External Interface Requirements**

**5.2.1 User Interfaces**

Assuming that data is present, a simple search bar will be given to user. Which then provides an output summary to the query in audio for the user.

**5.2.2 Hardware Interfaces**

* A system or a laptop, Speaker with mic.
* An internet connection (min. around 300kbps)

**5.2.3 Software Interfaces**

Table 3. Software interfaces

|  |  |
| --- | --- |
| **Software** | **Description** |
| Operating system | We have chosen windows operating system for its best support and user-friendliness. |
| Dataset | Available news data collected from various articles. |
| Applications | Jupyter notebook, Google chrome |
| API’s | The *IBM Watson* *Speech to Text* service provides *APIs* that use *IBM's speech*-*recognition.*  **gTTS** (*Google Text-to-Speech*). |

**5.2.4 Communication Interfaces**

The only communication is the final result that is the audio reading out the summary.

* 1. **System Features**

**5.3.1 Functional Requirements**

As mentioned at section 5.2, there are specific requirements which are detailed below.

First of all, the preparatory work, including the audio files, working environment, etc.

BERT, BART, GPT-2 models are considered.

The model selection is done by ROUGE metrics, which ever model provides best results for the selected data is selected for the summarizer.

* Text Summarizer Requirements:
* The system should provide text parser functions which can take the whole text and separate into sentences, paragraphs and words.
* The system should provide text-to-feature function which can take the necessary part and obtain a feature vector.
* The system should provide a well-trained Autoencoder to generate better inputs for classifier.
* The system should accept input in the form of audio/text
* The output generated will in the form of audio
* Three models namely, BERT, BART, GPT2 are considered.
* Single document summarization using BERT, BART, GPT2.
* Multi-document summarization using BERT, BART, GPT2.
* Audio summarization to summarize lengthy audio file.
* The model selection is done by ROUGE metrics, which ever model provides best results for the selected data is selected for the summarizer.

Collection of audio files and transcripting them using an API provided by IBM called SpeechToTextV1.The audio is recognized using a recognizer which uses a model called en-AU\_NarrowbandModel.

**5.3.2 Non-Functional Requirements**

**Usability:**

The system should be easy to use. The user should reach the summarized text with one button press if possible. Because one of the products main features is timesaving.

**Reliability:**

This software will be developed with machine learning, feature engineering and deep learning techniques. So, in this step there is no certain reliable percentage that is measurable

**Performance:**

As this project aims to save time of user, it should provide result as fast as possible.

**Supportability:**

The system should require Python knowledge for maintenance. If any problem occurs in server side and deep learning methods, it requires code knowledge

and deep learning background to solve.

**Portability:**

The application is built in such a way that it will be easy to transfer a software system transferred from its current hardware or software environment to another environment.

**Security:**

None, as we are not maintaining any customer account information.

**Maintainability:**

Th application is built in such a way that it is for a system to be supported, changed, enhanced, and restructured over time.

**Scalability:**

The system is scalable as it appropriately handles increasing (and decreasing) workloads.

**Chapter 6**

**DESIGN**

**6.1 Introduction**

Software design helps to transform user requirements into some suitable form, which helps the programmer in software coding and implementation. SRS document helps in assessing user requirements whereas for coding and implementation, there is a need of more specific and detailed requirements in software terms. The output of this process can directly be used into implementation in programming languages.

**6.2 Architecture Design**

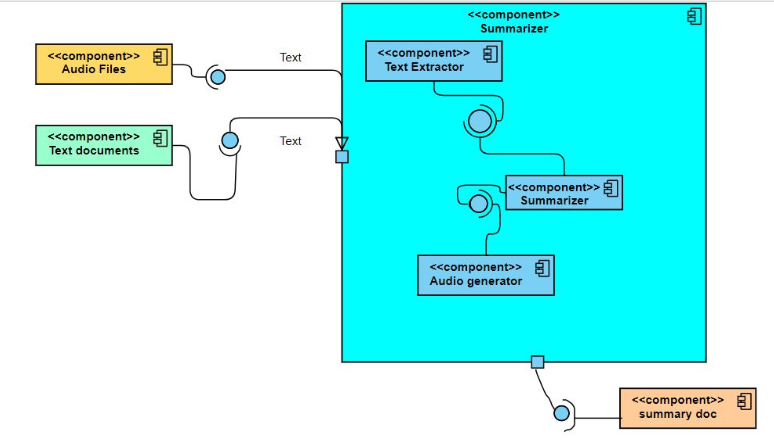


Fig 2.1 Architecture Design

Fig 2.1 represents the architectural design of our project.

* The input is taken in either audio or text format .If there are multiple text documents in the input then the next step is pre processing**.**The preprocessing step includes:
* Elimination of similar sentences by comparing all sentences of one document with another, and then check for the similarity between the sentences using cosine similarity and set a threshold to eliminate a particular sentence from that document if it crosses the threshold.
* After the elimination of all the similar sentences, all the articles are combined and is given as the single input to the summarizer
* Else if the input contains only one text document then there will be no pre processing step instead the next step will be the summarizer.
* Else if the input is in audio format then the next step is text extractor (IBM SpeechToTextV1)where audio is converted to text and is sent to summarizer
* The next step after generating summary is converting it into audio (GTTS (Google Text To Speech) API)which is delivered to final user.

**6.3 Graphical User Interface**

User interface is the front-end application view to which user interacts in order to use the software. The software becomes more popular if its user interface is:

* Attractive
* Simple to use
* Responsive in short time
* Clear to understand
* Consistent on all interface screens

In our application, we are using a GUI to enable users to interact with the application more effectively.

**6.4 Class Diagram and Classes**

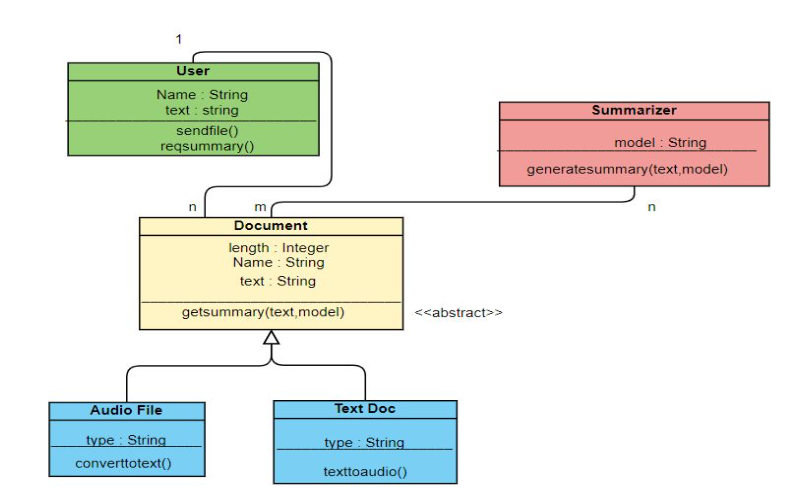


Fig 2.2 Class Diagram

**6.5 Sequence Diagram**

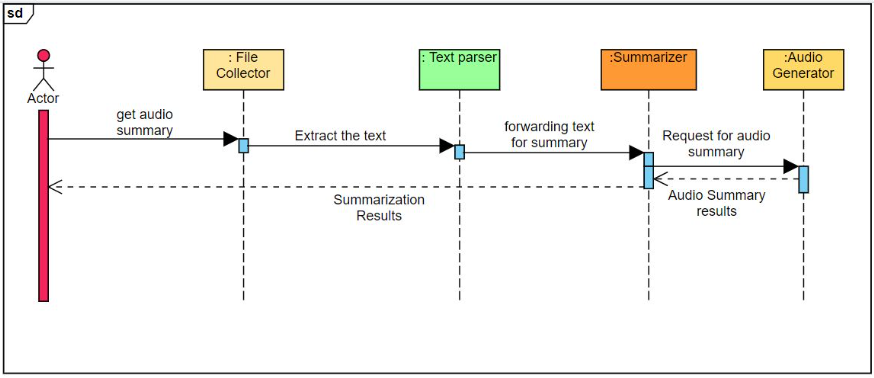


Fig 2.3 Sequence Diagram

**6.6 Data flow Diagram**

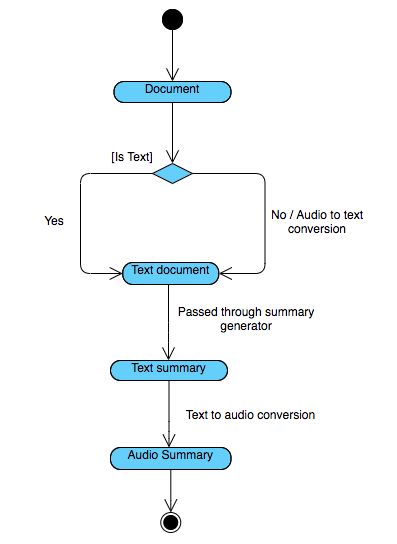


Fig 6.4 Data flow Diagram

**6.7 Conclusion**

The design helps in a better and clear implementation of the software. It provides a clear picture of control flow in the application. We have used Architecture design, GUI, Class diagrams, Sequence diagrams, Data flow diagrams to depict the system design. This can be used for further phases such as implementation.

**Chapter 7**

**IMPLEMENTATION**

* 1. **Technology Introduction**
* Software: Jupyter Notebook.
* Language: Python 3.
* Software method: We will use Agile methodology because it regularly reflects on how to become more effective.
  1. **Overall View of the Project in terms of Implementation**

**Multi Document Summarization:**

This process includes collection of articles on the same topic from different sources, it can be said that this process eliminates the need of reading different newspapers as we summarize collecting these articles from different news sources and provide you with the utmost important information. This was also extended for movie review summarization.

* **Pre-Processing:**

The preprocessing will include:

* Elimination of similar sentences: we compare a all sentences of one document with another, and then check for the similarity between the sentences using cosine similarity and set a threshold to eliminate a particular sentence from that document if it crosses the threshold.
* After the elimination of all the similar sentences, all the articles are combined and is given as the single input to the summarizer.
* **Summarization:**

This includes summarizing the preprocessed article using all 3 models BERT, BART, and GPT-2. The system generated summaries are then evaluated using the ROUGE metric which compares it with the human written summaries.

**Audio File Summarization:**

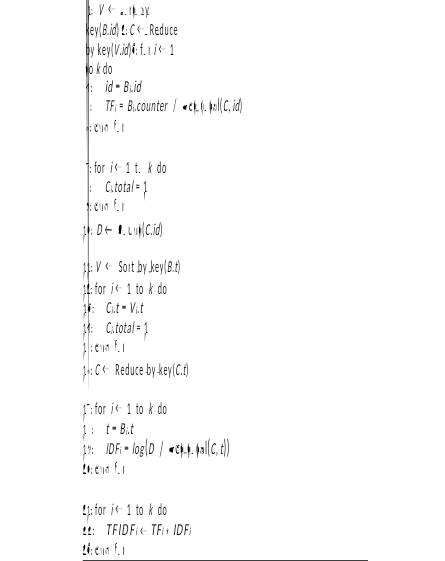
This process includes:

* Collection of audio files and transcripting them using an API provided by IBM called SpeechToTextV1. The audio is recognized using a recognizer which uses a model called en-AU\_NarrowbandModel.
* Preprocessing to form sentences from the transcript based on silences.
* Feeding the transcript to the summarizer.
* Evaluation of summary using ROUGE metric

**Audio Summaries:**

This process includes conversion of all the text summaries generated into audio format using GTTS (Google Text To Speech) API which takes in the text to the converted and desired language for the speech. This makes it easier for the blind to just listen to the summaries and helps in saving time as it can be listened from any place any time.

* 1. **Algorithm Pseudocode**
* **Multi Document Summarization:**
* Collecting articles on same topic from different sources
* Combine them as a single input to summarizer
* Summarize with BERT, GPT-2, BART
* Generate a concise summary
* **Audio files summarization:**
* Collecting the audio files.
* Transcribing them using python API’s
* Preprocessing for correcting grammar.
* Summarizing the transcribe.
* **Algorithm 1 Approximate TF-IDF (B,k)**



Algorithm 1 shows the pseudo-code for computing the approximate TF-IDF measure. The approximate TF-IDF measure is created on demand in buffer B and returns the k items (t; id) with the highest TF-IDF values in the given data stream. The algorithm has two phases. In the first phase, it computes A - tf and in the second phase A - idf. These values are used to compute A - tf - idf.

* **Pseudo Code for summarization:**

sort sentences by weight(tf-idf)

while (desiredSumLength is not met and there are unused sentences)

for (all sentences x )

if (sentence x not already in summary)

if (segment of sentece x has the lowest or equally low use)

set sum sentence = x

break out of for loop

end if

end if

end for

add sum sentence to the summary

record sum sentence as having been used

increment sum\_sentence's segment use

increment currentSumLength

end while

The pseudocode for the algorithm is shown in Figure 1. Because ideal compression levels (ratio of summary length to source length) for a document may change as users’ tasks change, flexibility to handle changes in summary length is built into the summarization algorithm. After sorting the sentences by weight and entering the while loop, in the pseudocode, the algorithm extracts a sentence from a topic segment if it has not already been used and if its text segment is the least represented in the summary. The highest-ranking sentence is always added to the summary first. This process continues until the summary contains the desired number of sentences or the document contains no unused sentences. Once the algorithm has extracted the desired number of sentences, it positions them in their original document order to promote fluency.

* 1. **Information on Implementation of Modules**
* NumPy: NumPy, which stands for Numerical Python, is a library consisting of multidimensional array objects and a collection of routines for processing those arrays. Using NumPy, mathematical and logical operations on arrays can be performed.
* TensorFlow: TensorFlow is an open-source machine learning framework. It is used for implementing machine learning and deep learning applications. TensorFlow is designed in Python programming language.
* Re: Python has a built-in package called re, which can be used to work with Regular Expressions. A RegEx, or Regular Expression, is a sequence of characters that forms a search pattern. RegEx can be used to check if a string contains the specified search pattern.
* Nltk: NLTK (Natural Language Toolkit) is a suite that contains libraries and programs for statistical language processing. It is one of the most powerful NLP libraries, which contains packages to make machines understand human language and reply to it with an appropriate response.
* Torch:Torch is an open-source machine learning library, a scientific computing framework. It provides a wide range of algorithms for deep learning and has an underlying [C](https://en.wikipedia.org/wiki/C_(programming_language)) implementation.
* Sklearn: Scikit-learn is a free machine learning library for Python. It features various algorithms like support vector machine, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.
* Pandas:Pandas is an open-source library in Python. It provides ready to use high-performance data structures and data analysis tools. Pandas module runs on top of NumPy and it is popularly used for data science and data analytics.

**Chapter 8**

**TESTING**

**8.1 Introduction**

Testing involves running the code with different inputs and tracking the performance of the application. Testing of the summarization tool is done to ensure the proper and flawless performance of the application.

**8.2 Test cases**

|  |  |
| --- | --- |
| Test Case ID | Test1 |
| Test Description | Single document summary using BERT. |
| Input | The document containing one news topic. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =52% |
| Test Result | PASS |

|  |  |
| --- | --- |
| Test Case ID | Test2 |
| Test Description | Single document summary using BART. |
| Input | The document containing one news topic. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =42% |
| Test Result | FAIL |

|  |  |
| --- | --- |
| Test Case ID | Test3 |
| Test Description | Single document summary using GPT-2. |
| Input | The document containing one news topic. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =55% |
| Test Result | PASS |

|  |  |
| --- | --- |
| Test Case ID | Test4 |
| Test Description | Multi-document summary using BERT. |
| Input | The document containing 3 news topics. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =50% |
| Test Result | PASS |

|  |  |
| --- | --- |
| Test Case ID | Test5 |
| Test Description | Multi-document summary using BART. |
| Input | The document containing three news topics. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =40% |
| Test Result | FAIL |

|  |  |
| --- | --- |
| Test Case ID | Test6 |
| Test Description | Single document summary using GPT-2 |
| Input | The document containing three news topics. |
| Expected Output | Summary with a rouge-1 score >45% |
| Actual Output | Summary with a rouge-1 score =52% |
| Test Result | PASS |

|  |  |
| --- | --- |
| Test Case ID | Test7 |
| Test Description | Audio input |
| Input | News articles are fed in audio format. (using GPT-2) |
| Expected Output | Summary in audio format. |
| Actual Output | Summary in audio format. |
| Test Result | PASS |

**Chapter 9**

**RESULTS & PERFORMANCE ANALYSIS**

**9.1 Result Snapshots**



Fig 9.1.1 New input article for GPT-2 (single document summarizer)

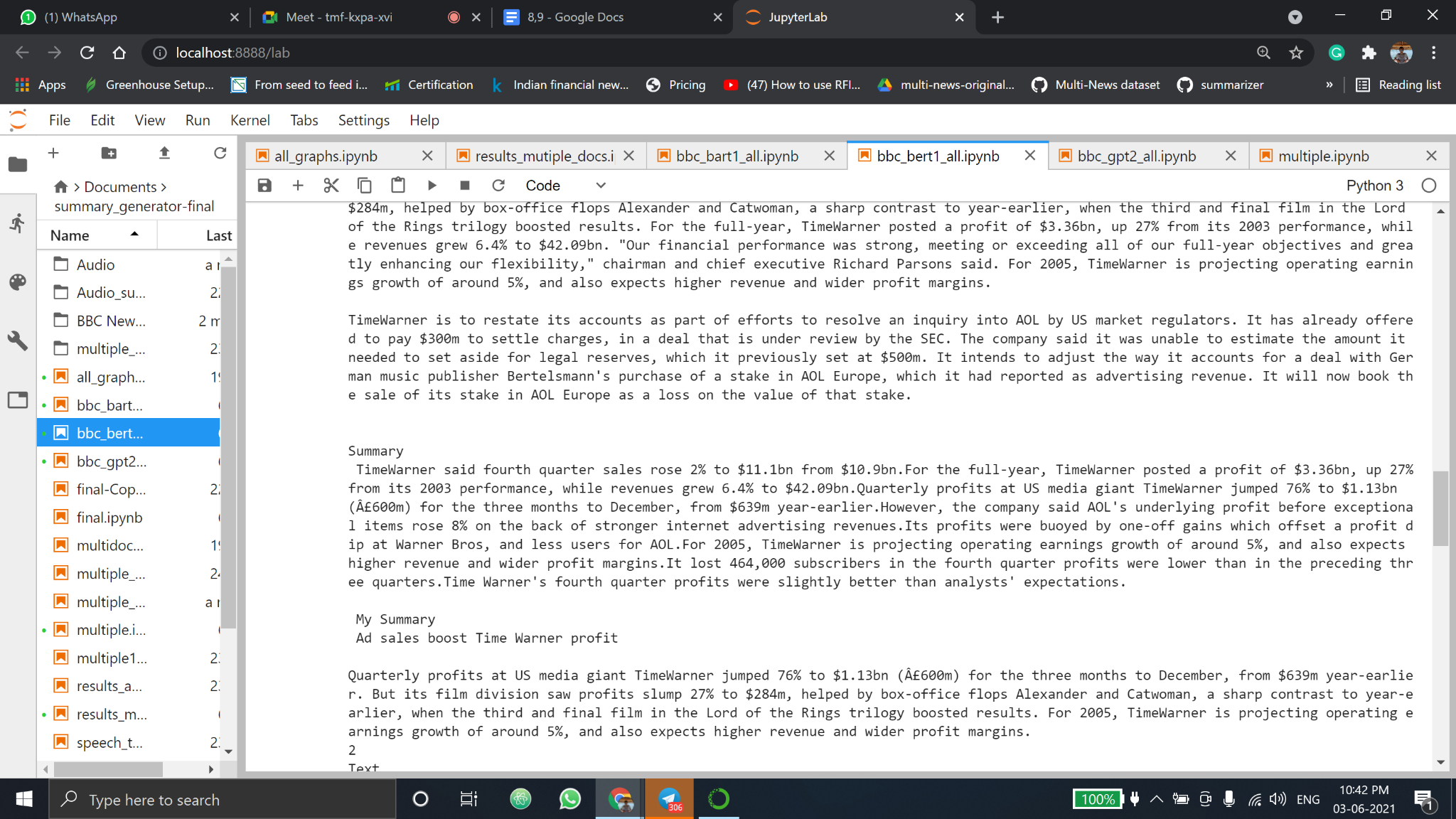


Fig 9.1.2 Expected Summary (Human-generated)

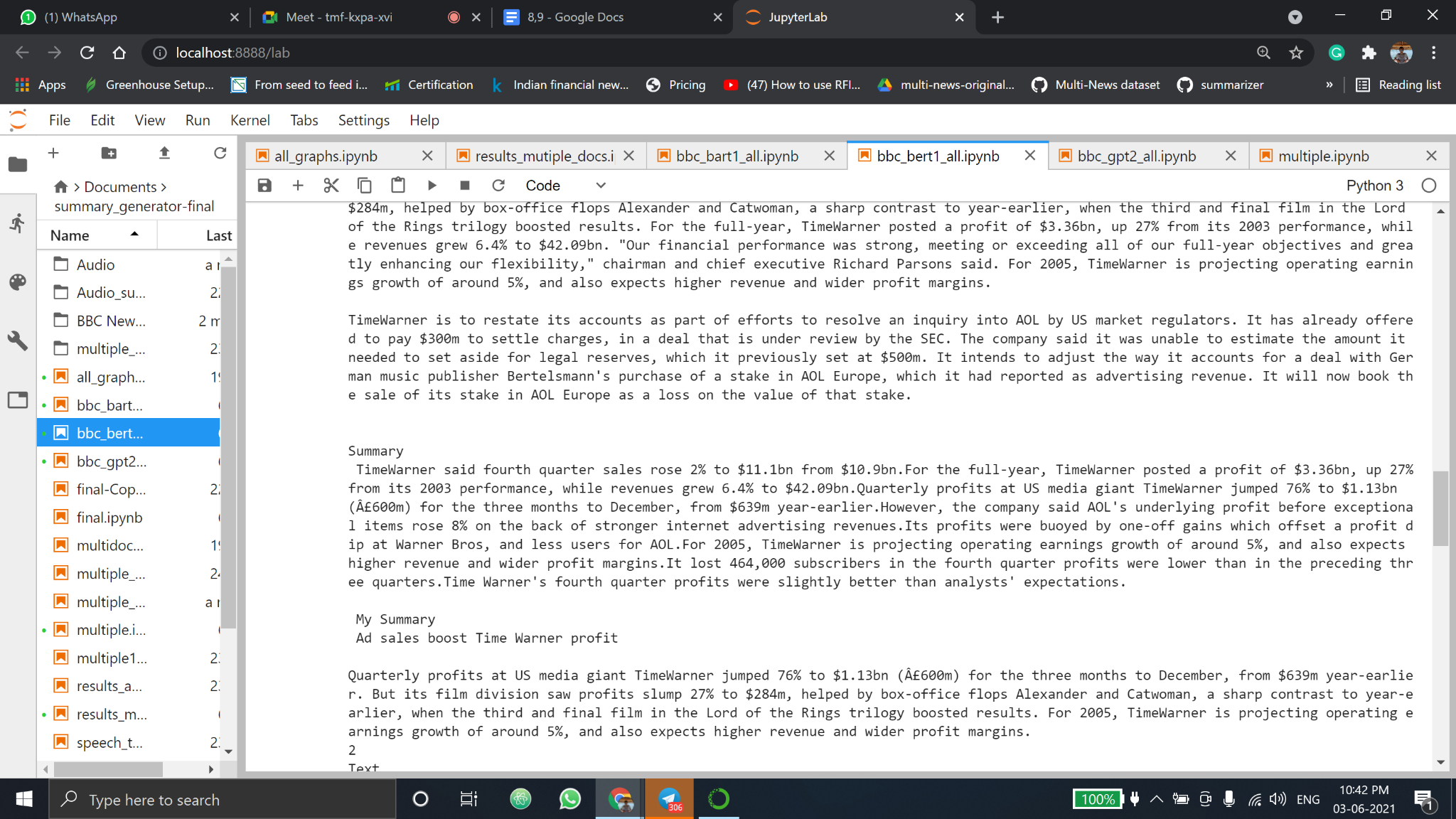


Fig 9.1.3 GPT-2 Summary (Computer-generated)

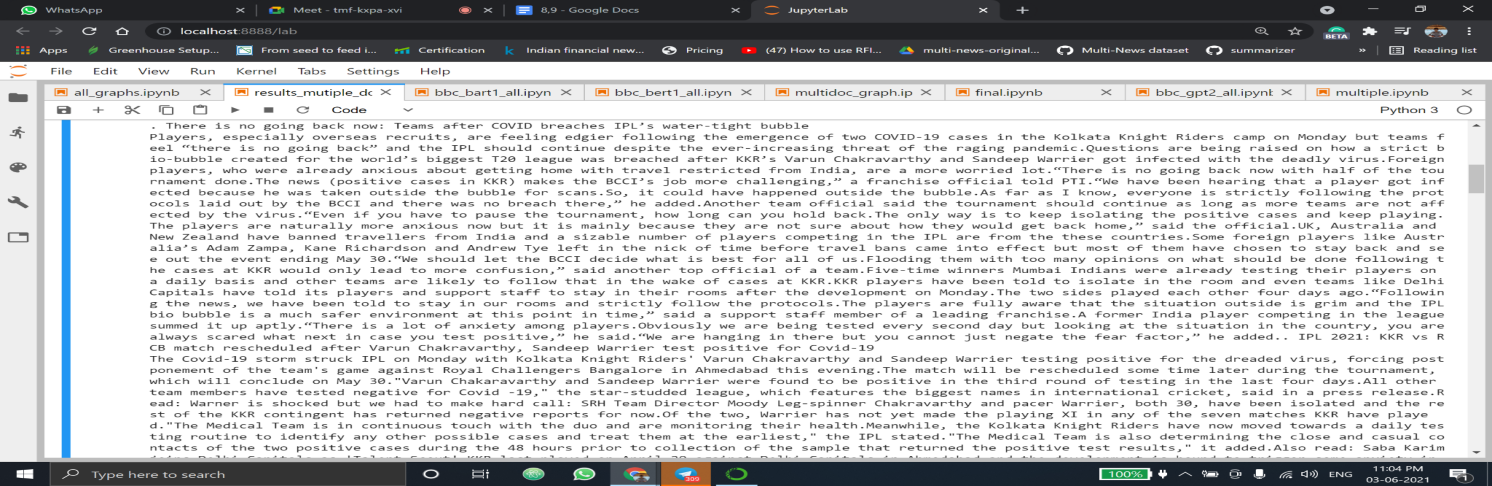


Fig 9.1.4 Multi-Document summarization: Document-1

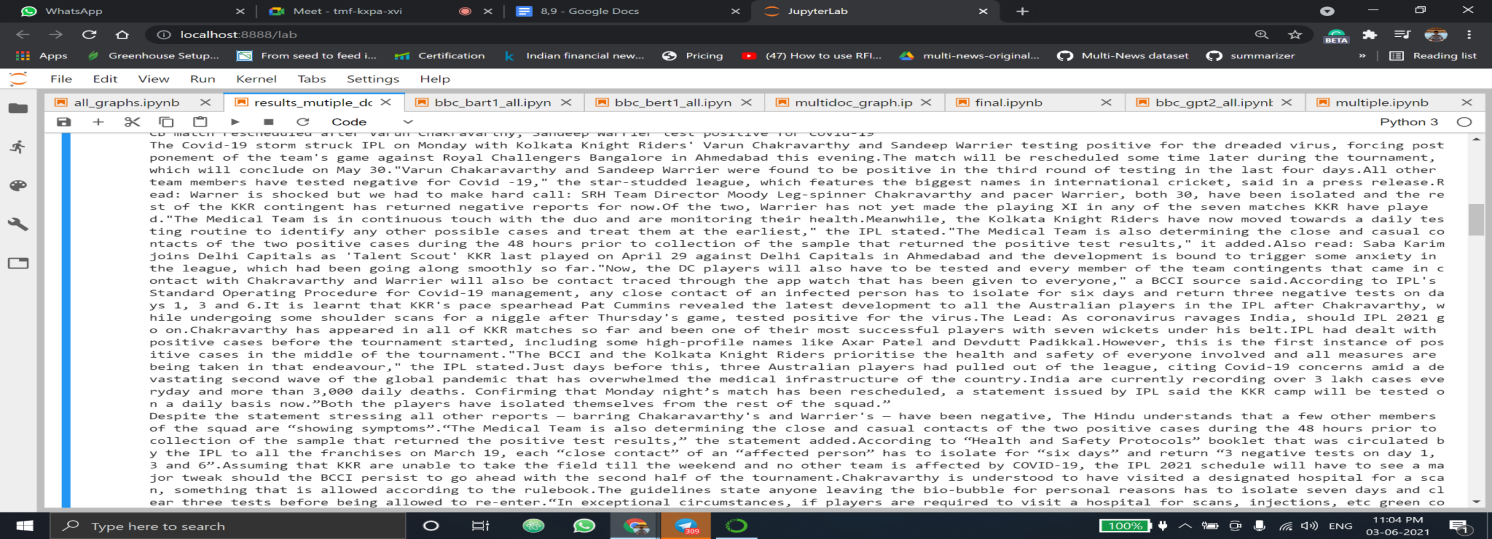


Fig 9.1.5 Multi-Document summarization: Document-2

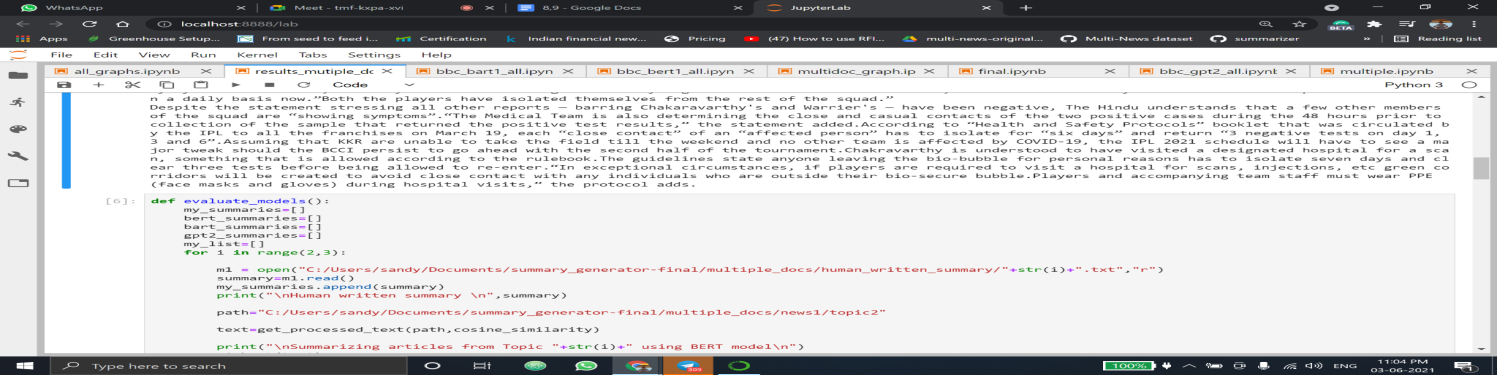


Fig 9.1.6 Multi-Document summarization: Document-3



Fig 9.1.7 Multi-Document summarization: Human written summary



Fig 9.1.8 Multi-Document summarization: Computer generated summary

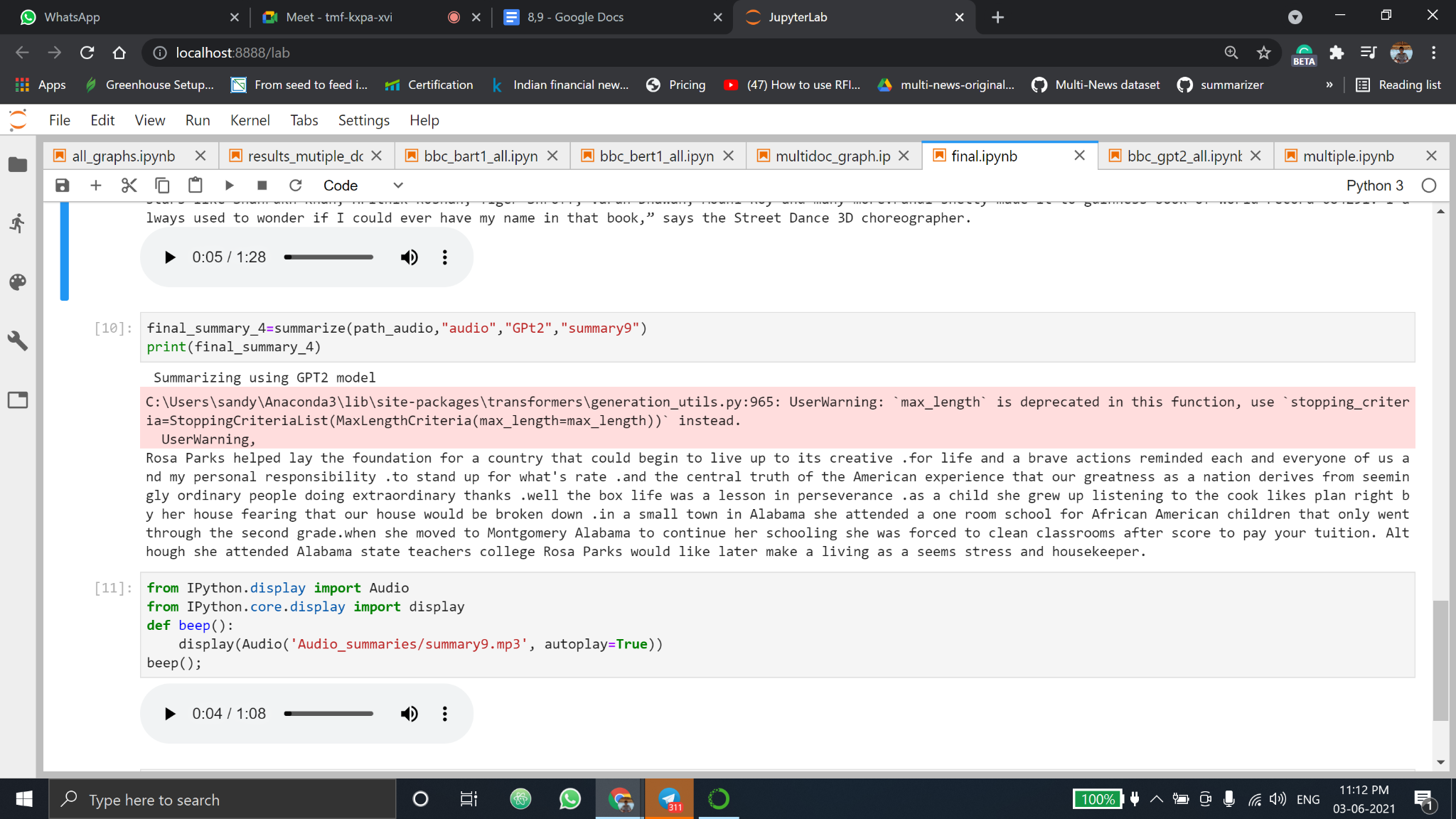
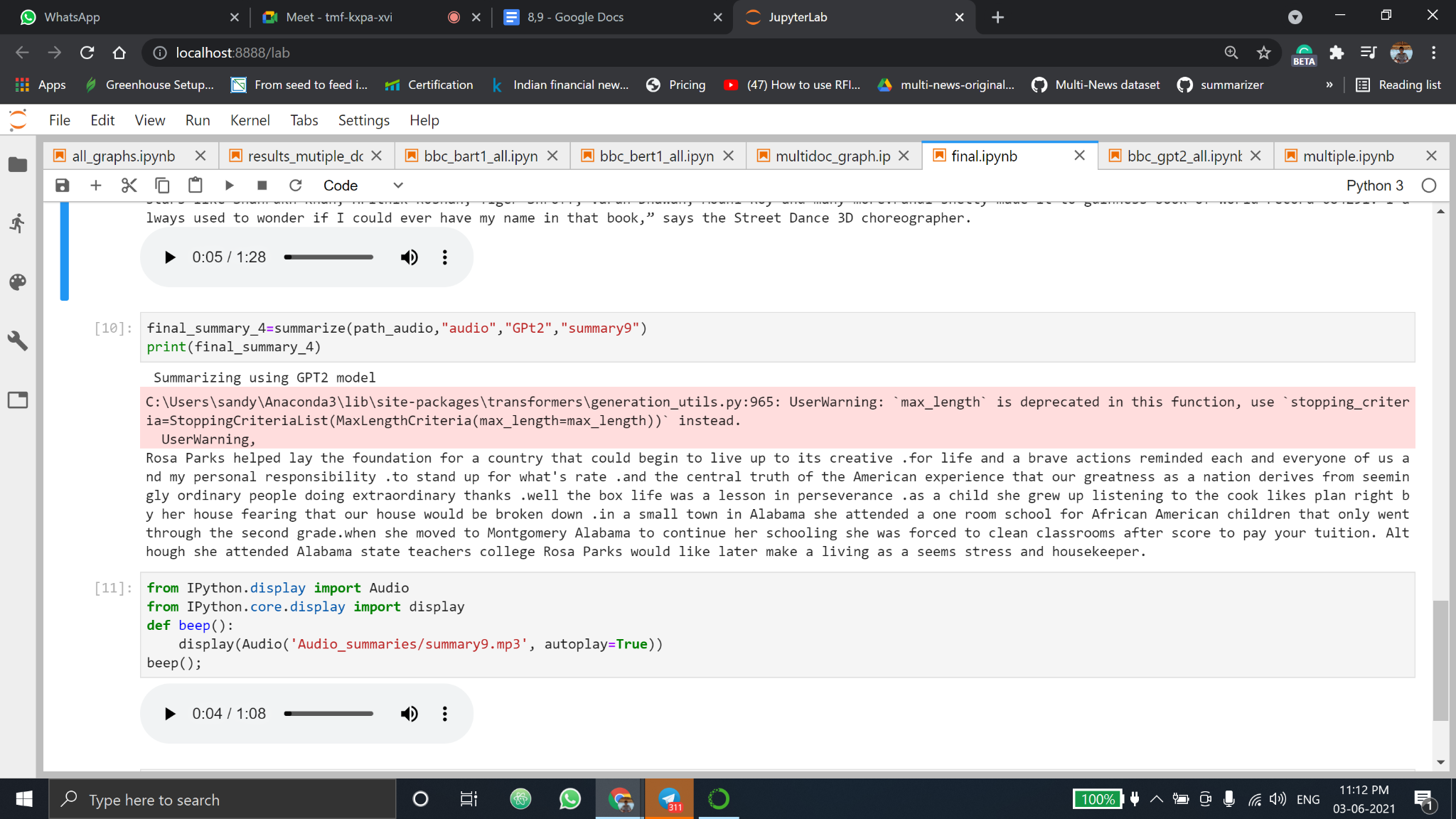


Fig 9.1.9 Audio as input news and generated audio summary

**9.2 Comparison Result Tables:**

Table 9.2.1 Single Document Summarization

|  |  |  |
| --- | --- | --- |
| Serial Number | Model Name | Rouge-1 Score |
| 1 | BERT | 52 |
| 2 | BART | 42 |
| 3 | GPT-2 | 55 |

Table 9.2.2 Multi-Document Summarization

|  |  |  |
| --- | --- | --- |
| Serial Number | Model Name | Rouge-1 Score |
| 1 | BERT | 50 |
| 2 | BART | 40 |
| 3 | GPT-2 | 52 |

Table 9.2.3 News with different opinion vs normal news

|  |  |  |
| --- | --- | --- |
|  | Controversial news | Non-Controversial News |
| Accuracy | 45% | 52% |

Table 9.2.4 Comparison for 3 articles and input articles

|  |  |  |
| --- | --- | --- |
|  | GPT-2 (3 input articles) | GPT-2 (5 input articles) |
| Accuracy | 52% | 51% |

**9.3 Performance Analysis – Graphs**

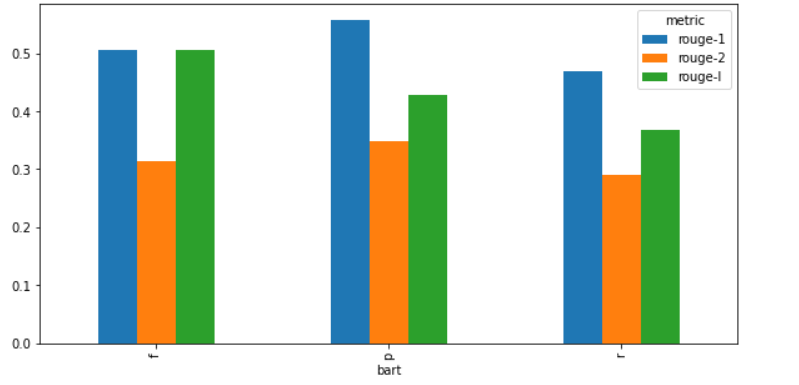


Fig 9.3.1 BART for single document summarization

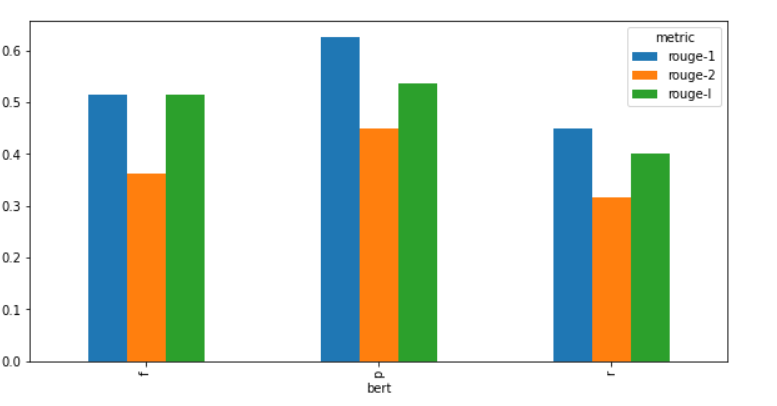


Fig 9.3.2 BERT for single document summarization

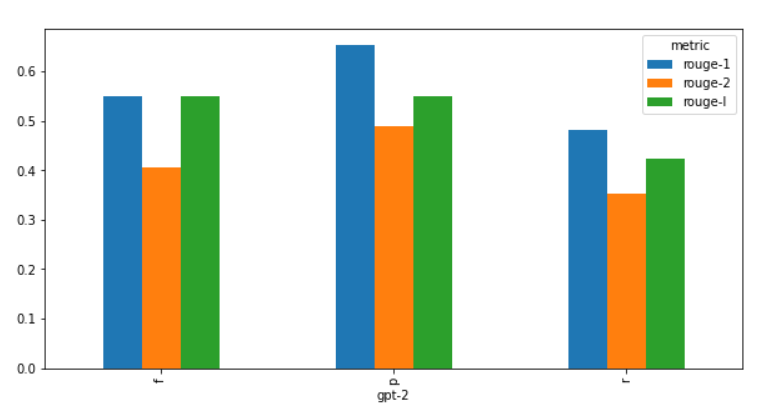


Fig 9.3.3 GPT-2 for single document summarization

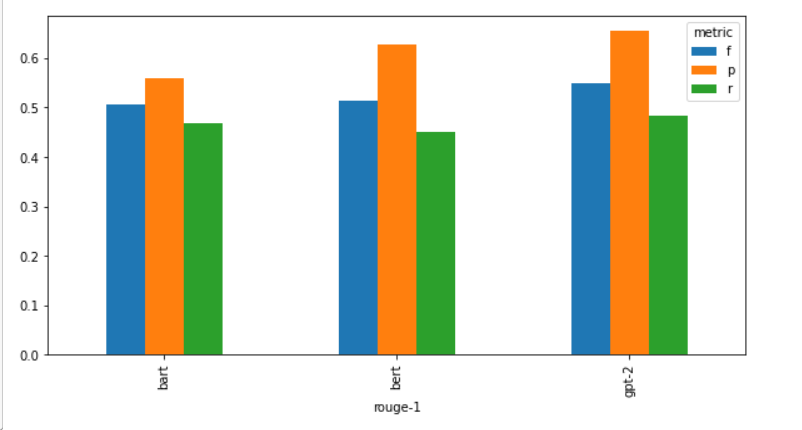


Fig 9.3.4 Comparison of BERT, BERT, GPT-2 for single document summarization

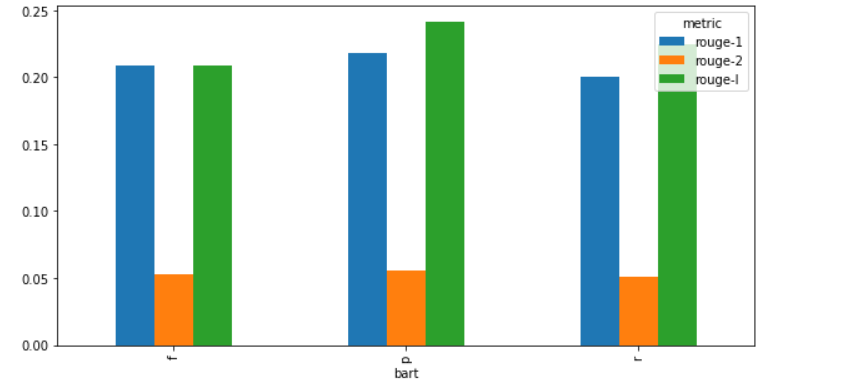


Fig 9.3.5 Rouge scores for Multiple document summarization using BART with Cosine Similarity

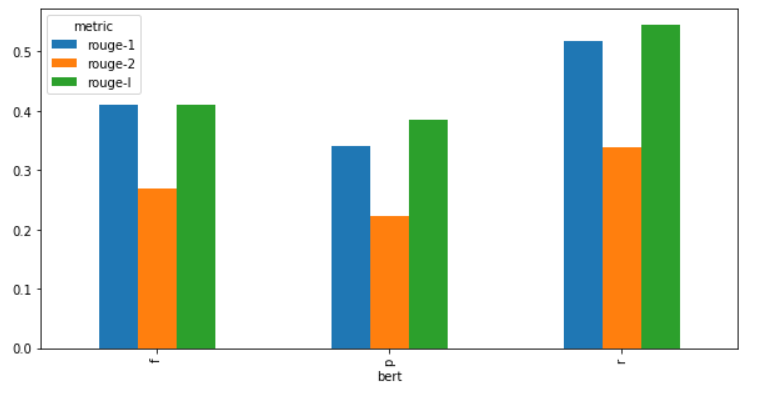


Fig 9.3.6 Rouge scores for Multiple document summarization using BERT with Cosine Similarity

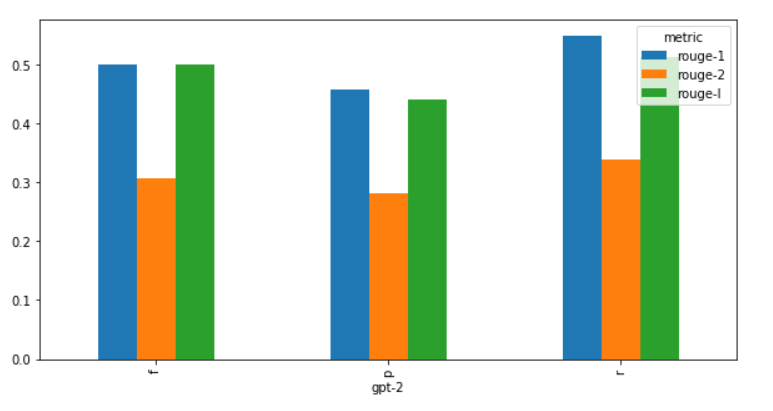


Fig 9.3.7 Rouge scores for Multiple document summarization using GPT-2 with Cosine Similarity

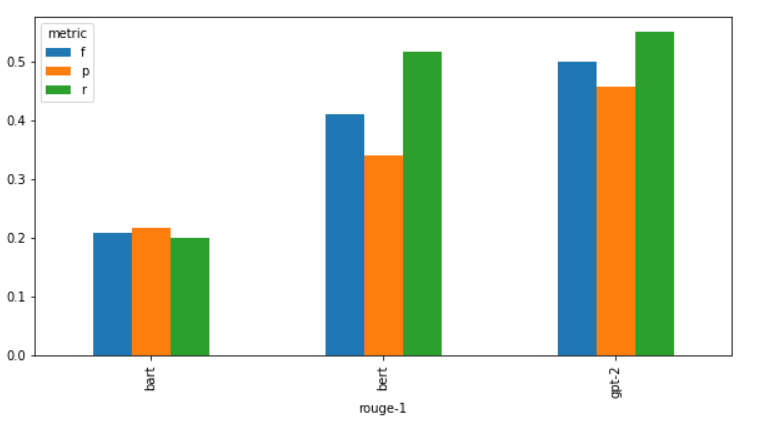


Fig 9.3.8 Rouge-1 scores for comparison of BERT, BART and GPT-2 in multi document summarization (Cosine similarity)

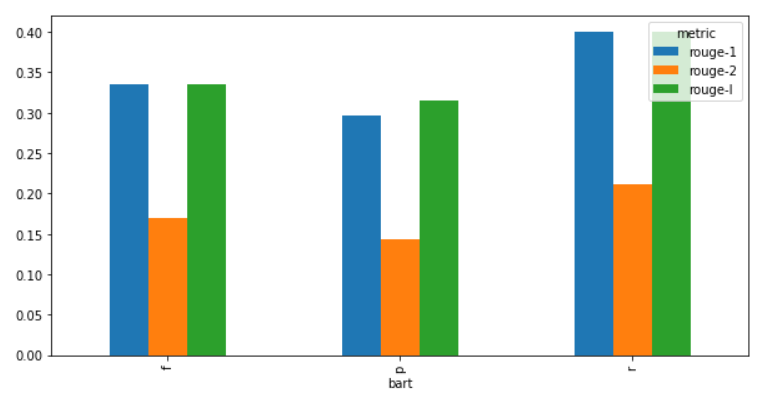


Fig 9.3.9 Rouge scores for Multiple document summarization using BART with Jaccard Similarity

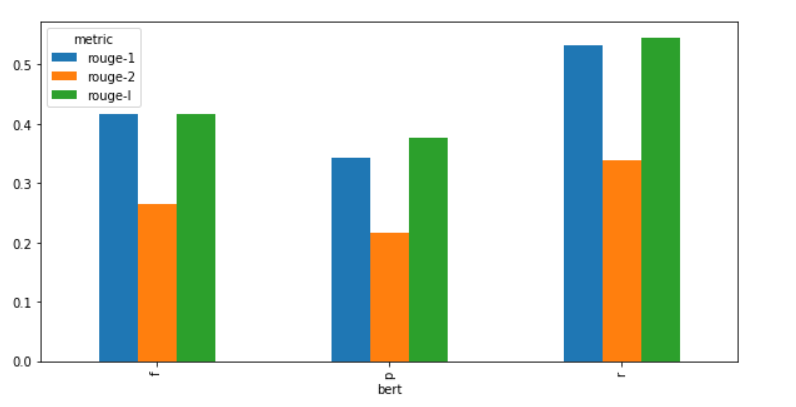


Fig 9.3.10 Rouge scores for Multiple document summarization using BERT with Jaccard Similarity

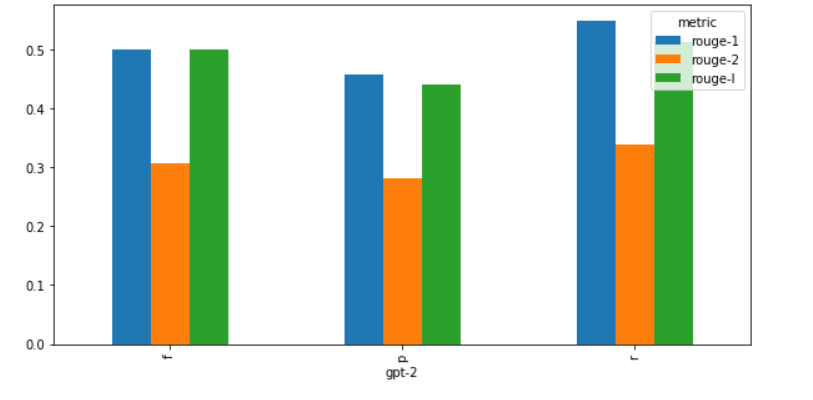


Fig 9.3.11 Rouge scores for Multiple document summarization using GPT-2 with Jaccard Similarity

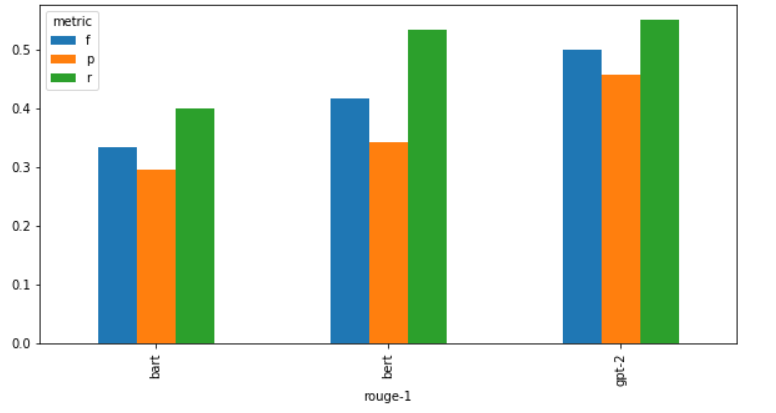


Fig 9.3.12 Rouge-1 scores for comparison of BERT, BART and GPT-2 in multi document summarization (Jaccard similarity)

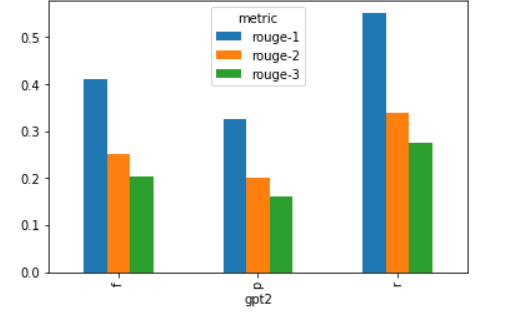


Fig 9.3.13 Rouge scores for mixed multi document summarization using GPT-2

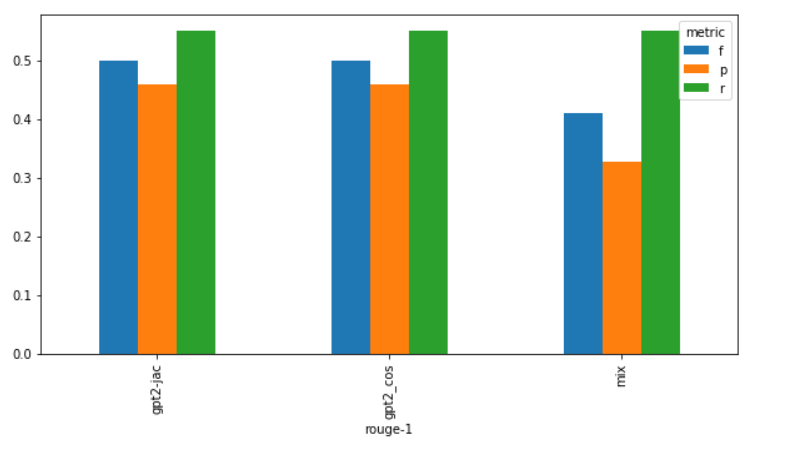


Fig 9.3.14 Rouge-1 scores for comparison of GPT-2 with cosine similarity, GPT-2 with Jaccard similarity and GPT-2 with mixed summarization

**Chapter 10**

**CONCLUSION & SCOPE FOR FUTURE WORK**

**10.1 Significance of the proposed research work**

* GPT-2 gives the best results for summarization when compared to BERT and BART in all the 5 domains because GPT-based model takes advantage of transfer learning
* The next best results are given by BERT
* The project aims to ease life of visually impaired through optimization and automation of technology for their benefit.
* The model summarizes the content, with help of NLP algorithms and that summarized content is finally converted to speech.
* This brings a major change in lives of the blind people, which is a step of success for us.
  1. **Directions for the future works**
* Since the results obtained from ROGUE metrics are highly dependent on the reference summaries, it cannot be considered as an absolute measure of the quality of a summary.
* However, among the available evaluation metrics, these are the most accurate.
* Future work might include better pre-processing of transcripts for audio file summarization.
* Bigger dataset can be built for multi-document summarisation
* Better audio summarisation

**REFERENCES**

[1] Extractive and abstractive summarization using BERT and GPT-2 on news and podcast data VICTOR RISNE ADÉLE SIITOVA

[2] Dimsum @LaySumm 20: BART-based Approach for Scientific Document Summarization

Yu, Tiezheng; Su, Dan; Dai, Wenliang; Fung, Pascale

[3] Wojciech Kry´sci´nski, Nitish Shirish Keskar and Bryan McCann, “Neural Text Summarization: A Critical Evaluation

[4] Yang Liu and Mirella Lapata, “Text Summarization with Pretrained Encoders”.

[5] Two-level\_text\_summarization\_from\_online\_news\_sources\_with\_sentiment\_analysis

[6] Single-Document\_Summarization\_Using\_Sentence\_Embeddings\_and\_K-Means\_Clustering

[7] https://www.sciencedirect.com/science/article/pii/S1319157820303712

[8] http://www.ajer.org/papers/v6(08)/ZE0608253260.pdf

[9] https://core.ac.uk/reader/190748390

[10] https://link.springer.com/chapter/10.1007/978-3-319-12640-1\_52