

“ End-End Summarization Tool using NLP ”

A Project Report Submitted to



**Rajiv Gandhi Proudyogiki Vishwavidyalaya
Towards Partial Fulfillment for the Award of Bachelor of
Engineering in *Information Technology***

Submitted by :-

Prayag Gavshinde(0827IT181065)

Raghav Agrawal(0827 IT 181068)

Rishabh Rathore(0827 IT 181073)

Rishi Somani(0827 IT 181074)

Sambhav Jain(0827IT181083)

Under Supervision of:

Prof. Kapil Sahu

(Associate Professor),

Information Technology

AITR, Indore



***Acropolis Institute of Technology & Research, Indore
Jan - June 2022***

EXAMINER APPROVAL

The Project entitled “ **End-End Summarization Tool using NLP**” submitted by Prayag Gavshinde(0827IT181065), Raghav Agarwal(0827IT181068),Rishi Somani(0827IT181074), Rishabh Rathore (0827IT181073), Sambhav Jain (0827IT181083), has been examined and is hereby approved towards partial fulfillment for the award of *Bachelor of Engineering degree in Information Technology* discipline, for which it has been submitted. It is understood that by this approval the undersigned does not necessarily endorse or approve any statement made, opinion expressed, or conclusion drawn therein, but approve the project only for the purpose for which it has been submitted.

(Internal Examiner)

(External Examiner)

GUIDE RECOMMENDATION

This is to certify that the work embodied in this project entitled “**End-End Summarization Tool using NLP**” submitted by **Prayag Gavshinde(0827IT181065),Raghav Agarwal(0827IT181068), Rishabh Rathore(0827IT181073),Rishi Somani(0827IT181074), Sambhav Jain(0827IT181083)**, have a satisfactory account of the bonafide work done under the supervision of **Prof. Kapil Sahu**, is recommended towards partial fulfillment for the award of the Bachelor of Engineering (Information Technology) degree by Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal.

(Project Guide)

(Project Coordinator)

STUDENTS UNDERTAKING

This is to certify that a project entitled “ **End-End Summarization Tool using NLP** ” has been developed by us under the supervision of **Prof. Kapil Sahu**. The whole responsibility of the work done in this project is ours. The sole intention of this work is only for practical learning and research. We further declare that to the best of our knowledge, this report does not contain any part of any work which has been submitted for the award of any degree either in this University or in any other University / Deemed University without proper citation and if the same work is found then We are liable for explanation to this.

Prayag Gavshinde(0827IT181065)

Raghav Agrawal(0827IT181068)

Rishabh Rathore(0827IT181073)

Rishi Somani(0827IT181074)

Sambhav Jain(0827IT181083)

Acknowledgment

We thank the almighty Lord for giving me the strength and courage to sail out through the tough and reach on shore safely.

There are several people without whom this project work would not have been feasible. Their high academic standards and personal integrity provided me with continuous guidance and support.

We owe a debt of sincere gratitude, a deep sense of reverence, and respect to our guide and mentor **Prof. Kapil Sahu**, AITR, Indore for his motivation, sagacious guidance, constant encouragement, vigilant supervision, and valuable critical appreciation throughout this project work, which helped us to complete the project on time.

We express profound gratitude and heartfelt thanks to **Prof. Prashant Lakkadwala**, HOD IT, AITR Indore for his support, suggestion, and inspiration for carrying out this project. I am very much thankful to other faculty and staff members of the IT Dept, AITR Indore for providing me with all support, help, and advice during the project. We would be failing in our duty if we did not acknowledge the support and guidance received from **Dr. SC Sharma**, Director, AITR, Indore whenever needed. We take the opportunity to convey my regards to the management of Acropolis Institute, Indore for extending academic and administrative support and providing me with all the necessary facilities for the project to achieve our objectives.

Executive Summary

“End-End Summarization Tool using NLP”

This project is submitted to Rajiv Gandhi Proudyogiki Vishwavidhyalaya, Bhopal(MP), India for partial fulfillment of Bachelor of Engineering in Information Technology branch under the sagacious guidance and vigilant supervision of **Prof. Kapil Sahu**.

The project is based on Deep Learning Learning, which is a subfield of Artificial intelligence, concerned with algorithms language(HTML)5, Cascading style sheet (CSS), bootstrap 4, jinja template for the development of the Front end.

List of Figures

1 System architecture for an abstractive approach	11
2 Sequence Diagram	12
3 Data flow Diagram.....	13
4 Workflow Diagrams.....	14
5 TEST CASES 1.....	35
6 TEST CASES 2.....	36
7 TEST CASES 3.....	37

List of Abbreviations

ML: Machine learning

DL: Deep learning

GPU: Graphics Processing Unit

CNN: Convolutional neural network

CSS: Cascading Style Sheets

HTML: Hypertext Markup Language

JS: Java Script

LDA: Latent Dirichlet allocation

BERT: Bidirectional Encoder Representations from
Transformers

T5 Transformer: Text To Text Transformer

Table of Contents

CHAPTER 1

INTRODUCTION	1
1.1 Overview	1
1.2 Background and Motivation	2
1.3 Problem Statement and Objectives.....	3
1.4 Scope of the Project.....	3
1.5 Team Organization.....	4

CHAPTER 2.

REVIEW OF LITERATURE.....	5
2.1 Preliminary Investigation.....	6
2.1. Existing System.....	7
2.2 Limitations of Existing System	8
2.3 Requirement Identification	8
2.3.1 Conclusion	9

CHAPTER 3.

3.Proposed System	9
3.1 The Proposal	9
3.2 Feasibility Study	9
3.3 Design Representation.....	10
3.3.1 Work Flow Diagrams.....	11
3.3.2 Sequence Diagrams.....	12
3.3.3 Data flow Diagram.....	13
3.3.4 ProcessDiagram	14
3.4 Data set	15

3.5 Requirements	15
3.5.1 Software	16
3.5.2 Hardware	16

CHAPTER 4.

IMPLEMENTATION	17
4.1 Technique Used	17
4.2 Tools Used	23
4.3 Language Used.....	30
4.4 Testing.....	33

CHAPTER 5 .

5.1 Conclusion.....	35
5.2 Limitations of the Work.....	36
5.3 Suggestion and Recommendations for Future Work.....	36

BIBLIOGRAPHY.....	37
--------------------------	-----------

ABSTRACT

Automatic text summarization is basically summarizing the given paragraph using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be useful. In this review, the main approaches to automatic text summarization are described. We review the different processes for summarization and describe the effectiveness and shortcomings of the different methods. Two types will be used i.e.- extractive approach and abstractive approach. The basic idea behind summarization is finding the subset of the data which contains the information of all the sets. There is a great need to reduce unnecessary data. It is very difficult to summarize the document manually so there is a great need for automatic methods. Approaches have been proposed inspired by the application of deep learning methods for automatic machine translation, specifically by framing the problem of text summarization as a sequence-to-sequence learning problem.

Chapter 1. Introduction

1.1 Overview

Automatic text summarization is the holy grail for people battling information overload, which becomes more and more acute over time. Hence it has attracted many researchers from diverse fields since the 1950s. However, it has remained a serious challenge, especially in the case of single news articles.

Text summarization is the process of generating a short, fluent, and most importantly accurate summary of a respectively longer text document. The main idea behind automatic text summarization is to be able to find a short subset of the most essential information from the entire set and present it in a human-readable format. As online textual data grows, automatic text summarization methods have the potential to be very helpful because more useful information can be read in a short time. The method of extracting these summaries from the original huge text without losing vital information is called **Text Summarization**. It is essential for the summary to be fluent, continuous, and depict the significance.

1.2 Background and Motivation

When you open news sites, articles, documents do you just start reading every news article? Probably not. We typically glance at the short news summary and then read more details if interested.

Short, informative summaries of the news are now everywhere like

magazines, news aggregator apps, research sites, etc. Well, It is possible to create the summaries automatically as the news comes in from various sources around the world.

In fact, google news, the inshorts app, and various other news aggregator apps take advantage of text summarization algorithms.

1.3 Problem Statement And Objective

The objective of the project is to understand the concepts of natural language processing and create a tool for text summarization. The concern in automatic summarization is increasing broadly so the manual work is removed. The project concentrates on creating a tool that automatically summarizes the document.

1. Automatic text summarizing by providing top sentences with the highest score in the document to save time.
2. Helping users to provide a proper caption or description for the image

1.4 Scope of the Project:-

1. It provides sensitivity to the client and adapts well to future summarization techniques.
2. It considers a complete document or article instead of fixed-length content.
3. It increases security and control.
4. It reduces IT Administration costs.

1.5 Team organization:-

- 1)**Prayag Gavshinde:** Along with a deep understanding of problem statements I have worked on the Backend part for understanding the techniques for the topic modeling part and applied, tested it on the backend.
- 2)**Raghav Agrawal:** Along with doing a preliminary investigation and limitations I worked on Main ML techniques for text summarization and image captioning. I have worked on a web-based framework Flask and database designing of MongoDB for deploying the Techniques on the frontend.
- 3)**Rishi Somani:** I investigated the project and helped in collecting the right dataset and I worked on the front end to design the result outcomes coming from the backend. Documentation and presentation are also a part where I made my contribution.
- 4)**Samhnav Jain:** Along with doing Preliminary investigation and understanding the needs of the project, I have worked in developing and aligning the front-end part. Documentation and presentation are also a part where I made my contribution.
- 5)**Rishabh Rathore:** I have studied the project requirements and helped to create Graphics designs for diagrams. Presentation is a major part of when I make my major contribution.

1.6 Report Structure :-

Chapter-1: Introduction - It gives the whole overview of the background and motivation of the problem and rationale to pick this project. The chapter describes the scope, objectives, and applications. Further, it provides the details of team members and their contribution To the development of the project.

Chapter 2: Review of Literature- explores the work done in the area of Project undertaken and discusses the limitations of existing systems and highlights the issues and challenges of the project area. The chapter finally ends up with the required identification for the present project work based on findings drawn from reviewed literature and end-user interactions.

Chapter 3: Proposed System - starts with the project proposal based on requirements identified, followed by benefits of the project. The chapter also illustrates the software engineering paradigm used along with different design representations. The chapter also includes a process flow diagram along with a block diagram and details of major modules of the project. The chapter also gives insights into different types of feasibility studies carried out for the

project undertaken.

Chapter 4: Implementation - Includes the details of different Technology/ Techniques/ Tools/ Programming Languages used in developing the Project. The chapter also includes the different user interface designs in the project and their functionality. Further, it discusses the experiment results along with testing of the project. The chapter ends with an evaluation of the project on different parameters like the accuracy and efficiency of various models.

Chapter 5: Conclusion - Concludes with objective-wise analysis of results and limitation of present work which is then followed by suggestions and recommendations for further improvement.

Chapter 2. Review of Literature

Automatic text summarization is basically summarizing the given paragraph using natural language processing and machine learning. There has been an explosion in the amount of text data from a variety of sources. This volume of text is an invaluable source of information and knowledge which needs to be effectively summarized to be useful. In this review, the main approaches and powerful Abstraction-based summarization method are described. We review the different processes for summarization of Images, text along with top topics document explains and describes the effectiveness and shortcomings of the different methods.

The basic idea behind summarization is finding the subset of the data which contains the information of all the sets. There is a great need to reduce unnecessary data. It is very difficult to summarize the document manually so there is a great need for automatic methods.

2.1 Preliminary Investigation

2.1.1 Existing systems

No .	Existing system/website/soft ware	Features Disadvantages	Limitations/ Gaps
1.	TLDRthis (website)	Summarize any article through URL or pasting text summarizing documents by uploading them from the local system is not possible.	cannot summarize pdfs. only applicable for articles or blogs through links.
2.	Resoomer(Website)	Summarize text for everyone By copy-paste content. Options for downloading possible. generated summary in pdf of any doctype.	cannot summarize pdfs. only applicable for text content.
3	addtext(Website)	AddText is written using the latest web technologies and works as well on your create your own memes, they do not make intuitive feature observations on objects or actions in the	automatically generating natural language descriptions according to the content observed in an image,

		create great looking posters	image,	
--	--	---------------------------------	--------	--

2.2 Limitations of Existing System

1. The summary is less accurate.
2. Time constraint is less.
3. Computational speed is more.
4. Cannot Summarize Pdfs, Only applicable for articles or blogs and text content which we upload.

2.3. Requirement Identification

3.1 FUNCTIONAL REQUIREMENTS:-

1. Large Chunks of Text
2. Web Page URLs
3. Text Rank Algorithm.
4. Word2vec Representation(Glove Algorithm)

3.2 Non FUNCTIONAL REQUIREMENTS:-

1. Reliability
2. Performance
3. Usability
4. Platform independent
5. Supportability.

2.4. Conclusion

This chapter reviews the literature surveys that have been done during the research work. The related work that has been proposed by many researchers has been discussed. The existing system does not focus on Advanced NLP Techniques. End-user does not get reliable summaries. The existing system uses the Abstractive Text Summarization Technique Which does not give a reliable summary.

Chapter 3. Proposed System

3.1 The Proposal

The solution we proposed is well suited for extracting summaries of each kind on a single interface where the user enjoys all the benefits along with a voice clone feature to listen as well as read.

The major functionality that the project enjoys are

1. **Document Summarization:** Any pdf, doc, or text files like research papers, chapter notes, or anything if you upload it will find the best sentences with the highest dominating score and generate a short and sweet summary.
2. **The article, Blog summary through URL:** If you want to read a summary of any online article available on any website, just provide a link. You will get important details like author, Published date, and article summary.
3. **Topic Modelling:** With the advent of big data and Machine Learning along with Natural Language Processing, it has become the need of an hour to extract a certain topic or a collection of topics that the document is about. Think when you have to analyze or go through thousands of documents and categorize under 10 – 15 buckets. How tedious and boring will it be? Thanks to Topic Modeling where instead of manually going through numerous documents, with the help of Natural Language Processing and Text Mining, each document can be categorized under a certain topic.
4. **Image Captioning:** Sometimes It feels tedious to describe any

Image. We will integrate a model which will describe and provide a perfect description of an Image.

3.2 Feasibility Study

A Feasibility Study is an analysis of how successfully a system can be implemented, technical, operational, and economical to determine its positive, negative outcomes before investing a considerable amount of time into it. The feasibility of the system has been studied from various aspects like whether the system is feasible technically operationally and economically.

3.2.1 Technical: The Technology Needed for the proposed system that we are going to develop is available. we can develop our system still using this technology If needed to upgrade. In the future, if we want to use this technology as an Android app for our system, It is possible. Hence, the system that we are going to develop will successfully satisfy the needs of the system for it to be technically feasible. The system, once set up completely, works fine and is capable of generating summary documents, Auto Generate Image Caption, Generates Topic to Documents.

3.2.2 Operational: The user interface will be user-friendly and no training will be required to use the application. The solution proposed for our project is operationally workable and most likely convenient to solve the problem of numerous users.

3.2.3 Economical: Since the system is developed as a part of project work, there is no manual cost to spend for the proposed system. Also, all the resources are already available. The cost to conduct a full system investigation is negotiable because required information is collected from the internet. Hence, the project that we are going to develop won't require an enormous amount of money to be

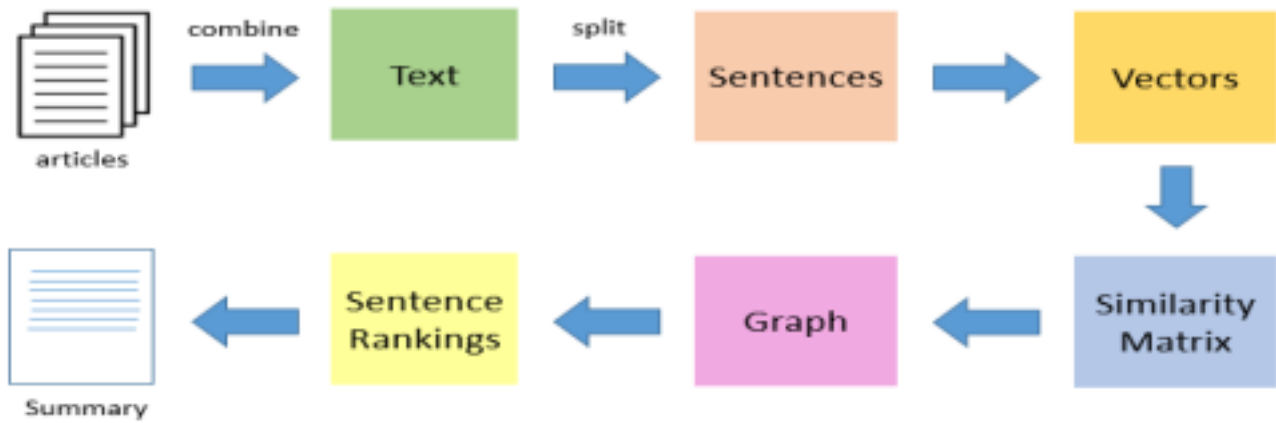
developed so it will be economically feasible.

3.3 Design Representation

3.3.1 System Architecture for Abstractive Approach



3.3.2 Sequence Diagram



3.3.3 Work Flow Diagram

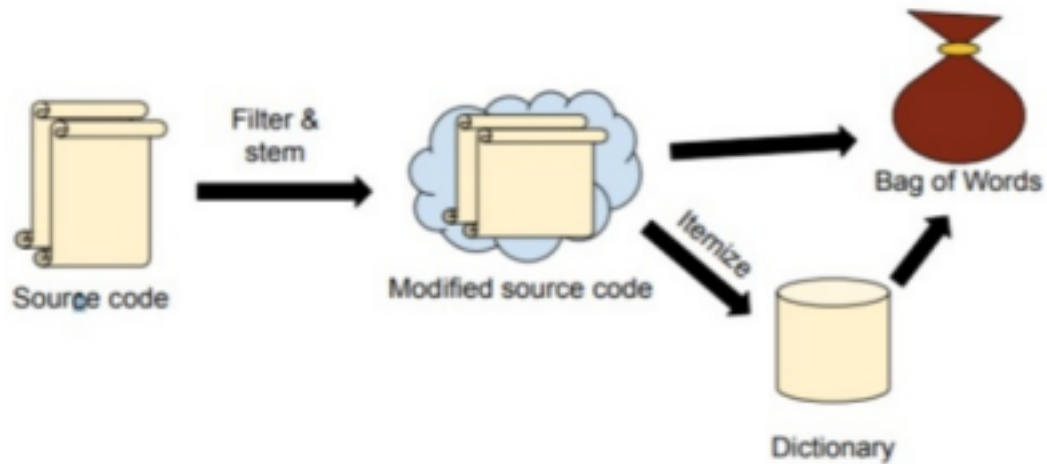
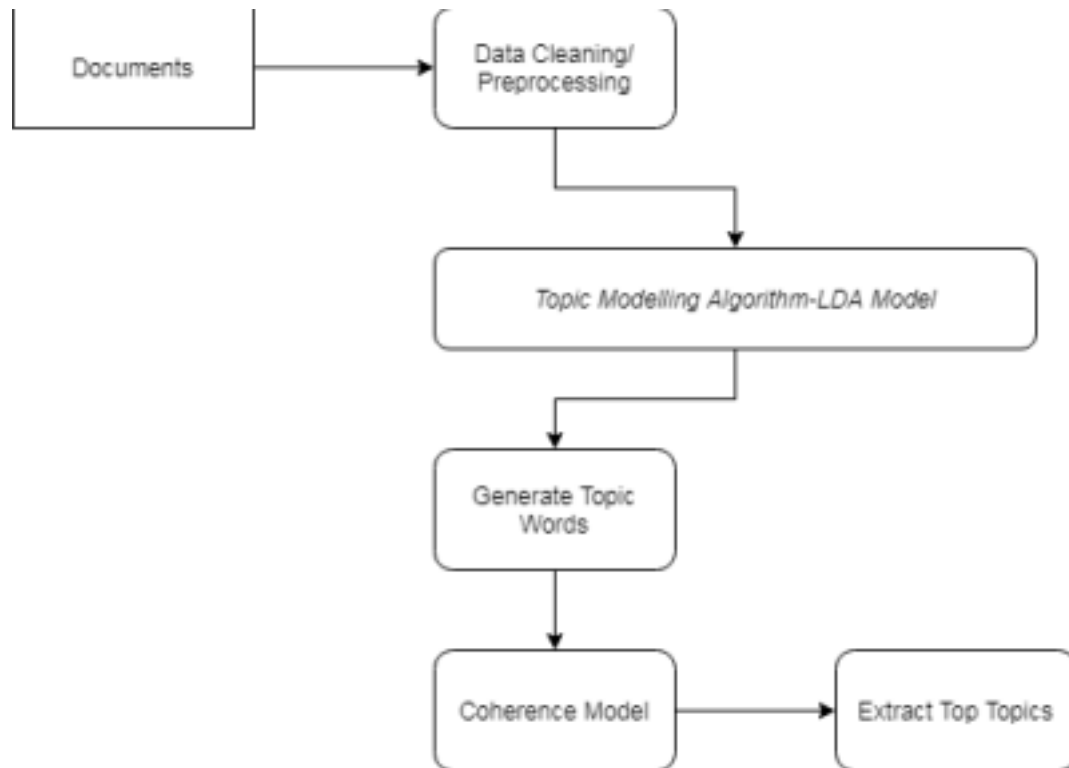


Fig. Workflow diagram for creating dictionary and BOW representation



3.4 Dataset

3.4.1 Data collection -

Flickr 8K dataset is the most popular Image captioning dataset which we collected from Kaggle. Kaggle is a data science learning, and competitive platform where you can learn, practice, participate in various data science competitions, and contribute your code, an idea to the community and explore your skills. The Flickr 8K dataset is used to train the Image captioning model which contains 8000 images and a separate file describing captions of each image.

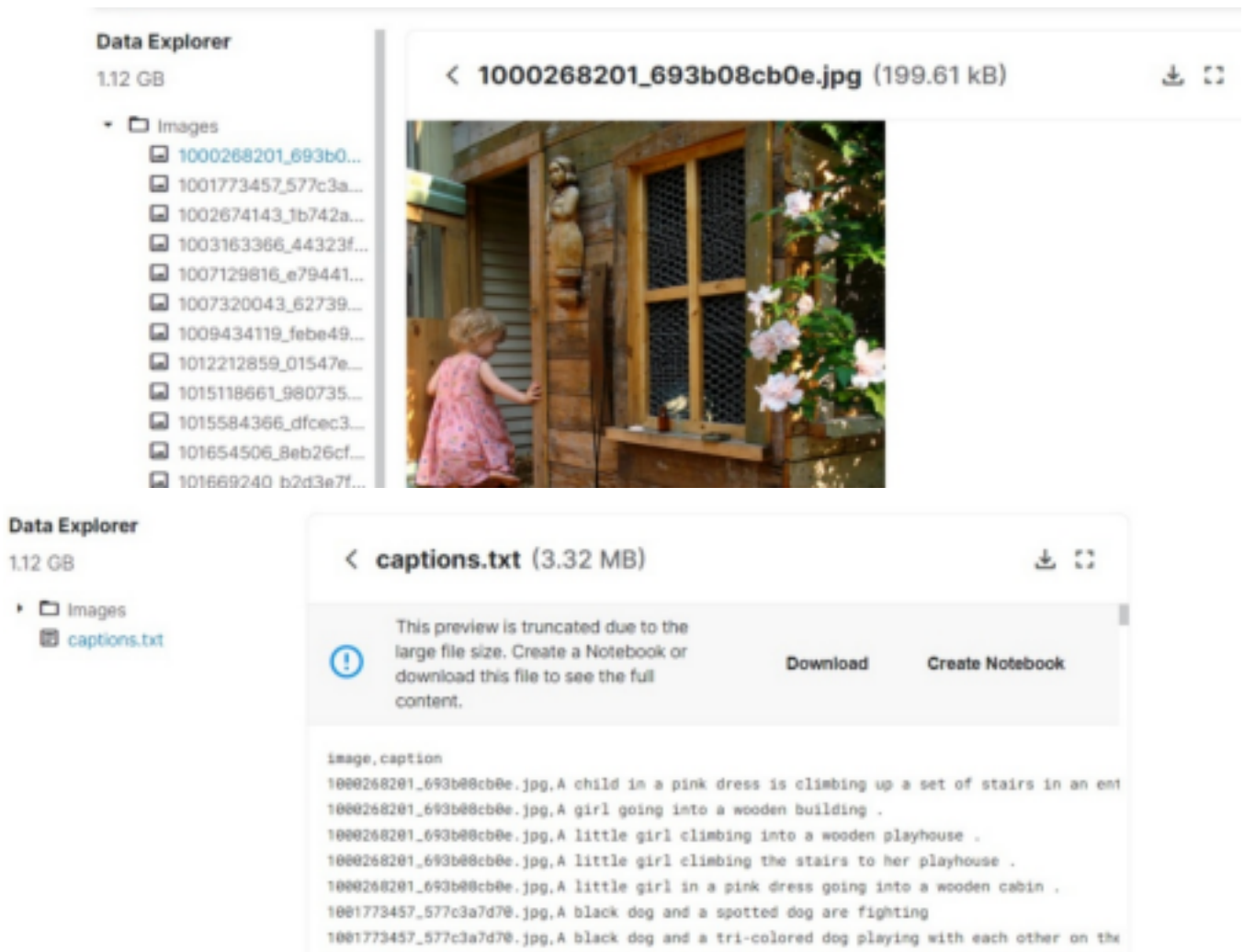


Fig. Image Captions File

Topic Modeling and Text summarization are achieved using Transformer Models so various documents, articles, text paragraphs were picked from different websites on google to validate and test for correct parameters to pass to Transformer.

3.5 REQUIREMENT:-

3.5.1 Software-Python IDE and its libraries,Jupyter-Notebook,Google Colab,VScode.

3.5.2 Hardware-

❖ Modern Operating System:

☐ Windows 7 or 10

☐ Mac OS X 10.11 or higher, 64-bit

☐ Linux: RHEL 6/7, 64-bit (almost all libraries also work in Ubuntu) ❖ 4 GB RAM

❖ x86 64-bit CPU (Intel / AMD architecture)

Chapter-4. Implementation

4.1 Techniques Used

4.1.1 Deep Learning

Deep Learning is a subfield of machine learning concerned with algorithms inspired by the structure and function of the brain called artificial neural networks. Deep learning (also known as deep structured learning or hierarchical learning) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms. Learning can be supervised, semi-supervised or unsupervised.

4.1.2 Convolutional Neural Networks(CNNs)

For solving the problem of Image and its feature extraction, we take the help of Convolutional Neural Networks (CNN) as they are proven to be very powerful in the field of computer vision. A CNN is a class of deep neural networks widely used in image classification, object detection, and all other computer vision tasks. CNN is inspired by the connectivity pattern of the neurons in the human and animal visual cortex. The 3 reasons why CNN is preferred is that they do not require much rendering of the images whereas; traditional image recognition requires lots of rendering before getting the result.

4.1.3 Natural Language Processing

Natural Language Processing (NLP) is a field in Computer Science that focuses on the study of the interaction between human

languages and computers (Chowdhury, 2003). Text summarization is in this field because computers are required to understand what humans have written and produce human-readable outputs. NLP can also be seen as a study of Artificial Intelligence (AI). Therefore many existing AI algorithms and methods, including neural network models, are also used for solving NLP-related problems. With the existing research, researchers generally rely on two types of approaches for text summarization: extractive summarization and abstractive summarization (Dalal and Malik, 2013).

4.2 Models Developed

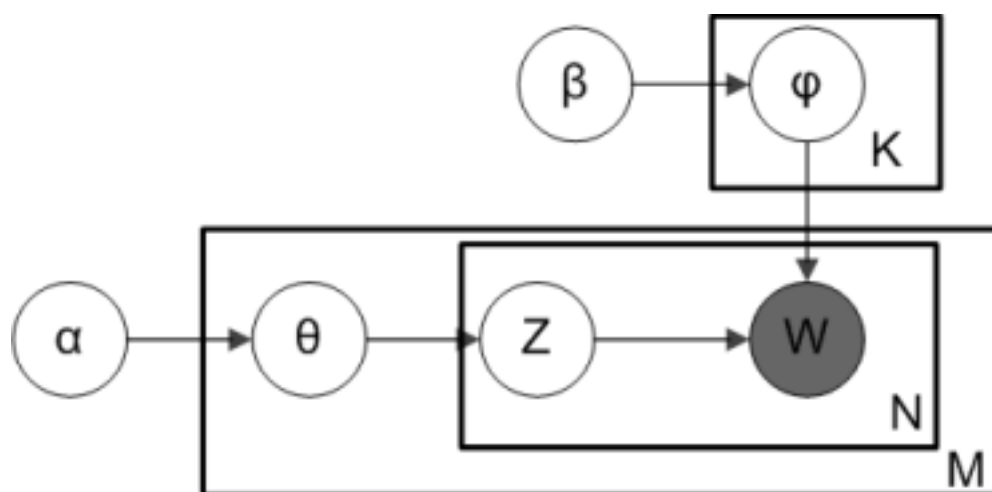
4.2.1 LDA(Latent Dirichlet allocation) For Topic Modelling

LDA is a form of unsupervised learning that views documents as bags of words. LDA works by first making a key assumption: the way a document was generated was by picking a set of topics and then for each topic picking a set of words.

1. Assume there are k topics across all of the documents
2. Distribute these k topics across document m (this distribution is known as α and can be symmetric or asymmetric, more on this later) by assigning each word a topic.
3. For each word w in document m , assume its topic is wrong but every other word is assigned the correct topic.
4. Probabilistically assign word w a topic based on two things:
 - what topics are in document m
 - how many times *has* word w has been assigned a particular topic across all of the documents (this distribution is called β , more on this later)

5. Repeat this process a number of times for each document and you're done!

The Model



1. α is the per-document topic distribution,
2. β is the per-topic word distribution,
3. θ is the topic distribution for document m ,
4. ϕ is the word distribution for topic k ,
5. z is the topic for the n -th word in document m , and
6. w is the specific word

4.2.2 Abstraction Based summarization using T5 Transformer

In abstraction-based summarization, advanced deep learning techniques are applied to paraphrase and shorten the original document, just like humans do. Think of it as a pen that Google each produces novel sentences that may not be part of the source document. Since abstractive machine learning algorithms can generate new phrases and sentences that represent the most important information from the source text,

they can assist in overcoming the grammatical inaccuracies of the extraction techniques.

T5 in many ways is one of its kind transformers architecture that not only gives state-of-the-art results in many NLP tasks but also has a very radical approach to NLP tasks. T5 uses common crawl web extracted text. The authors apply some pretty simple heuristic filtering. T5 removes any lines that didn't end in a terminal punctuation mark. It also removes lines with the word javascript and any pages that had a curly bracket (since it often appears in code). It deduplicates the dataset by taking a sliding window of 3 sentence chunks and deduplicates it so that only one of them appears in the dataset. For example, above 3 pages, the last paragraph on the middle page is removed since the same content appears on the first page. It ends up with 750 gigabytes of clean-ish English text.

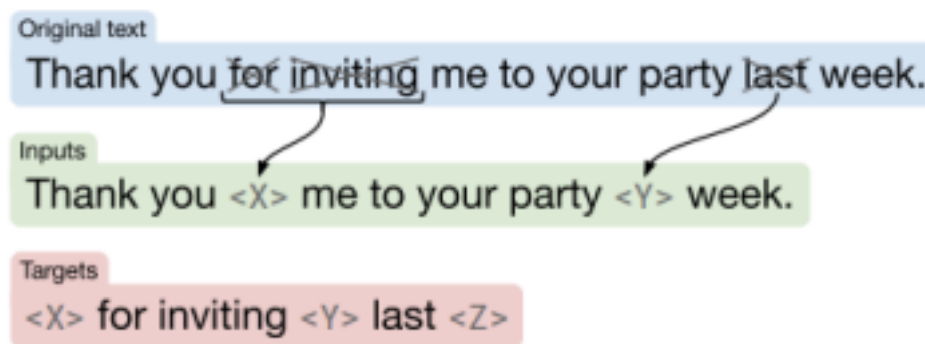


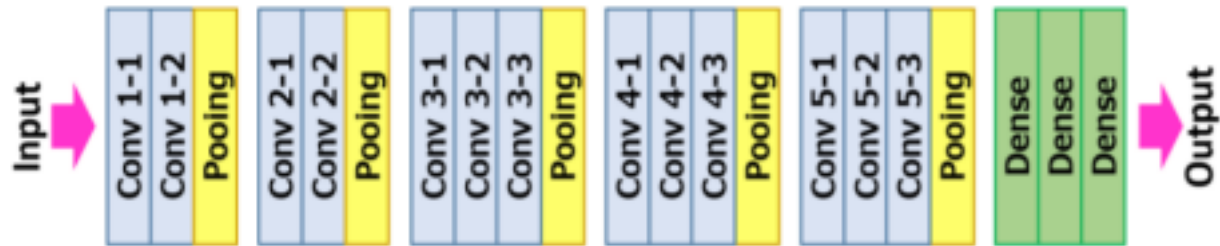
Fig. Unsupervised Objective T5 Transformer

4.2.3 Images Captioning Using Autoencoders(Vgg16)

VGG- Network is a convolutional neural network model proposed by K. Simonyan and A. Zisserman in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". This architecture achieved top-5 test accuracy of 92.7% in ImageNet, which has over 14 million images belonging to 1000 classes.

It is one of the famous architectures in the deep learning field. Replacing large

kernel-sized filters with 11 and 5 in the first and second layer respectively showed improvement over AlexNet architecture, with multiple 3×3 kernel-sized filters one after another.



4.3 ALGORITHM SPECIFICATION

INPUT:- Large Chunks of Text/articles/Web Page URLs

OUTPUT:- Summarized Text and Audio File of Summarized Text

STEP1:- Concatenate all the contained in the articles

STEP2:- Entire concatenated Text is Split into individual Sentences.

STEP3:- Find Vector Representation(Word Embeddings) for each and every sentence by using Glove Algorithm.

STEP4:- Similarities Between sentence vectors are calculated and stored in a matrix using Cosine Similarity

STEP5:- Convert the similarity matrix into Graph using Page Rank Algorithm.

STEP6:- Find a certain number of top-ranked sentences using the page rank algorithm to form a summary.

STEP7:- Convert the summarized text into an audio file using Google Text To Speech API.

4.4 Tools Used

1. Gensim:

Gensim = “*Generate Similarly*” is a popular open-source natural language processing (NLP) library used for unsupervised topic modeling. Gensim, implemented in Python and Cython, is designed to handle large text collections using data streaming as well as incremental online algorithms. This makes it different from those machine learning software packages that target only in-memory processing.

2. Keras:

Keras is the high-level API of TensorFlow 2.0: an approachable, highly-productive interface for solving machine learning problems, with a focus on modern deep learning. It provides essential abstractions and building blocks for developing and shipping machine learning solutions with high iteration velocity. Keras empowers engineers and researchers to take full advantage of the scalability and cross-platform capabilities of TensorFlow 2.0: you can run Keras on TPU or on large clusters of GPUs, and you can export your Keras models to run in the browser or on a mobile device.

3. Numpy:

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, various derived objects (such as masked arrays and matrices), and an assortment of routines for fast operations on arrays, including mathematical, logical, shape manipulation, sorting, selecting, I/O, discrete Fourier

transforms basic linear algebra, basic statistical operations, random simulation and much more.

4.Transformers: Transformers provides thousands of pretrained models to perform tasks on texts such as classification, information extraction, question answering, summarization, translation, text generation, and more in over 100 languages. It aims to make cutting-edge NLP easier to use for everyone.

4.5 Languages Used

4.4.1 Backend: Python

Python is used as a primary language because Python is a general-purpose, dynamic, high-level, and interpreted programming language. It supports an Object-Oriented programming approach to develop applications. It is simple and easy to learn and provides lots of high-level data structures. And also due to many reasons such as:

- 1) Cross-platform Language: Python can run equally on different platforms such as Windows, Linux, UNIX, and Macintosh, etc. So, we can say that Python is a portable language.
- 2) Extensible: It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our
- 3) Python code. It converts the program into bytecode, and any platform can use that byte

code.

- 4) Large Standard Library: It provides a vast range of libraries for various fields such as machine learning, web development, and also for scripting. There are various machine learning libraries, such as Tensor flow, Numpy, Keras, and Gensim.
- 5) Dynamic Memory Allocation: In Python, we don't need to specify the data type of the variable. When we assign some value to the variable, it automatically allocates the memory to the variable at run time.

4.4.2 Framework: Flask

Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Jinja2

Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages.

Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have a built-in abstraction layer for database handling, nor does it have validation support. Instead, Flask supports the extensions to add such functionality to the application. Some of the popular Flask extensions are

discussed later in the tutorial.

4.4.3 Frontend: Bootstrap

- 1) Bootstrap is a free front-end framework for faster and easier web development.
- 2) Bootstrap includes HTML and CSS-based design templates for typography, forms, buttons, tables, navigation, modals, image carousels, and many others, as well as optional JavaScript plugins.
- 3) Bootstrap also gives you the ability to easily create responsive designs

HTML:

- 1) HTML stands for HyperText Markup Language.
- 2) HTML is the standard markup language for creating Web pages → HTML describes the structure of a Web page

CSS:

- 1) CSS stands for Cascading Style Sheets
- 2) CSS describes how HTML elements are to be displayed on a screen, paper, or in other media

4.5 Testing

Text Summarizing Outputs

Input Article	Original Summary (Title)	Generated Summary (Title)

<p>File names are appearing in the attachment details page twice and blank file names are observed. PFA screenshot</p>	<p>File names are appearing in the attachment details page twice also blank file names observed</p>	<p>blank file names are appearing in the attachment details page twice. also blank file names are observed in the attachment details page.</p>
--	---	--

<p>So the HC Policy has to be defined as Check the process is running Check the file with MD5 checksum Hence IVE administrator would have to define both the Host Checker policies and implement at the role or realm level. One workaround is to combine Process Checking and File MD5 Checksum. This issue applies until IVE version 6.</p>	<p>Host Checker fails with " Can't Generate Checksum" error [Applicable until IVE version 6.1rx]</p>	<p>IVE administrator would have to define both the Host Checker policies and implement at the role or realm level. one workaround is to combine Process Checking and File MD5 Checksum.</p>
<p>M160 FRUs Hot-swappable Routing Engine Switching and Forward... Requires Power Down Circuit Breaker Box Connector Interface Panel CIP M160 Chassis includes backplane Please</p>	<p>[Archive] What are the specific Field Replaceable Units (FRU) on the M160?</p>	<p>M160 Chassis includes a backplane. requires a power-down circuit breaker box.</p>

refer to the M160 hardware guide for more detail.

Topic Modeling Results

```
def Extract_topics(text, n_topics=8):
    text = preprocessing_text(text) #txt cleaning
    lst_of_tokens = [text.split()] #word tokenize
    dict_ = corpora.Dictionary(lst_of_tokens)
    doc_term_matrix = [dict_.doc2bow(i) for i in lst_of_tokens]
    lda = gensim.models.ldamodel.LdaModel
    ldamodel = Lda(doc_term_matrix, num_topics=6, id2word = dict_, passes=15, random_state=0, eval_every=None)
    topics = ldamodel.print_topics(num_topics=1, num_words=n_topics)
    all_topics = {}
    for top in topics:
        s = top[1]
        lst = s.split('+')
        for i in lst:
            word = i.split("'")[1]
            word = re.sub("'", '', word)
            word = re.sub(' +', '', word)
            all_topics.add(word)
    return all_topics
```

```
text = "Alpha and Beta Hyperparameters - alpha represents document-topic density and Beta represents topic-word density. High"
topics = Extract_topics(text, num_topics)
topics
```

```
{'contain',
 'corpus',
 'documenttopic',
 'fewer',
 'hand',
 'hyperparameters',
 'large',
 'number',
 'topicword',
 'value'}
```

Chapter 5. Conclusion

5.1 Conclusion

With the ever-growing text data, text summarization seems to have the potential for reducing the reading time by showing summaries of the text documents that capture the key points in the original documents. Automatic text summarization is an old challenge but the current research direction diverts towards emerging trends in biomedicine, product review, education domains, emails, and blogs. This is due to the fact that there is information overload in these areas, especially on the World Wide Web. Automated summarization is an important area in NLP (Natural Language Processing) research. It consists of automatically creating a summary of one or more texts. The purpose of extractive document summarization is to automatically select several indicative sentences, passages, or paragraphs from the original document. Text summarization approaches based on NLP have, to an extent, succeeded in making an effective summary of a document. Both extractive and abstractive methods have been researched. Most summarization techniques are based on extractive methods. As with time the internet is growing at a very fast rate and with it data and information are also increasing. It will be difficult for humans to summarize large amounts of data. Thus there is a need for automatic text summarization because of this huge amount of data.

5.2 Limitations of the Work

1. Require Stable Internet Connection as it's working on the web.
2. Multiple documents or Multiple Images cannot be summarized at a

time.

3. Image captioning Model trained on a dataset of most commercial and used Images. There is a chance to not generate captions on rare Images.

5.3 Suggestion and Recommendations for Future Work

We have implemented Automatic text summarization using an abstractive method. Further, after using RNN and LSTM the accuracy is still very low for a summarizer. Furthermore, we will be using machine learning for semantic text summarization for more accurate summaries and will try to make a grader which will grade the document according to English grammar. There are many text summarizers available but all do not give appropriate results. Thus we will be using machine learning algorithms to increase the effectiveness of the automatic summarizer.

Bibliography

- [1] G. Erkan, D. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization", Journal of Artificial Intelligence Research 22, 2004.
- [2] Selvani Deepthi Kavala, Dr. Radhika Y, "Extractive Text Summarization Using Modified Sighing and Sentence Symmetric Feature Methods", I.J. Modern Education and Computer Science, 2015.
- [3] H.P.Luhn, "The Automatic Creation of Literature Abstracts". IBM Journal of Research and Development, 1958.
- [4] H.P. Edmundson, "New Methods in Automatic Extracting", Journal of the Association for Computing Machinery, April 1969.
- [5] A.Das, M.Marko, A.Probst, M.A.Portal, C.Gershenson —Neural Net Model For Featured Word Extraction, 2002.
- [6] Jagadeesh J, Prasad Pingali, Vasudeva Varma, "Sentence Extraction based single Document Summarization", Workshop on Document Summarization, 19th and 20th March 2005, IIIT Allahabad. [7] Arman Kiani B, M. R. Akbarzadeh —Automatic Text Summarization Using: Hybrid Fuzzy GA-GP, IEEE International Conference on Fuzzy Systems. July 16-21, 2006.
- [8] R. Mihalcea, and P. Tarau, "Textrank: Bringing order into texts,". In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004.
- [9] R. Nallapati, B. Zhou, C. dos Santos, C. Gulcehre, and B. Xiang, "Abstractive text summarization using sequence-to-sequence RNNs and beyond,". In Computational Natural Language Learning, 2016