**S.P. Mandali’s**

**Ramnarain Ruia Autonomous College**

**Matunga, Mumbai - 400019**

**Department of Computer Science & Information Technology**

**Project Report**

**On**

**Summarify**

**(Text Summarization using Extractive Methods)**

**Project Guide**

**Prof. Rasika Mundhe**

**Project By**

**Ms. Trushali Chawda**

**MSC Computer Science**

**2023-24**



# Acknowledgement

I would like to express my heartfelt gratitude to my project guide Prof. Rasika Mundhe, for providing guidance and support throughout my project completion. Her knowledge and feedback have helped me learn new concepts and explore possible solutions for my project, "Summarify," which is a Text Summarization using Extractive Methods.

I would also like to give my humble gratitude to Prof. Megha Sawant and the entire teaching and non-teaching staff of the CSIT department for giving me this opportunity.

Lastly, a special thanks to my parents and friends for supporting me throughout the project development process.

Sincere Thanks from,

Trushali Chawda

**INDEX**

|  |  |  |
| --- | --- | --- |
| **SR. NO.** | **PARTICULARS** | **PAGE NO.** |
| 1 | Abstract | 5 |
| 2 | Introduction | 6 |
| 3 | Literature Review | 7 |
| 4 | Research Methodology | 8 |
| 5 | Experimental setup | 12 |
| 6 | Results | 13 |
| 7 | Conclusion | 32 |
| 8 | References | 32 |

# Text Summarization using Extractive Methods

## 1. Abstract

The Internet is overloaded with a vast amount of information that comes from various academic websites, social media, blogs, and articles. For a user who seeks to find relevant information from various topics in a short span of time, going through the entire document can be time-consuming and inefficient. To address this challenge, a document summary can be generated, which extracts only the relevant information from the text. This will not only reduce the time but will also provide the exact information for which the user is looking. This project proposed various extractive-based approaches to provide better summaries, which will help users to learn about the important context of the article.

## 2. Introduction

Currently, on the internet nearly more than 4 million TB of data is produced every day. When users put their query on the internet and search for the results, they are flooded with sites providing a lot of information. Some of the articles may provide the same content or depict the same information in a different context. This makes the search of the user time-consuming also sometimes it is unable to get relevant information to their queries. To provide a solution to this challenge, a text summarization approach is used.

Text Summarization is the process of condensing a longer piece of text into a shorter version while retaining its main ideas and key points. The goal is to extract the most important information from the original text and present it in a concise and coherent manner. It helps in getting quick summaries of any news websites, articles or blogs.

Text Summarization is mainly divided into three types: extractive based, abstractive based and hybrid based each will be explained in detail.

**Extractive Based Text Summarization** - It involves selecting and presenting existing important sentences or phrases from the document into a summary. It does not generate a new sentence. Statistical algorithms, graph-based methods and machine learning techniques are used to identify most important words. It preserves the original wording and maintains coherency. The disadvantage is it may not create well-structured summaries especially for longer text. Redundancy is also possible in multi-document summarization.

**Abstractive Based Text Summarization -** It involves generating new sentences that convey the main ideas from the original text. It generates new words to create summaries, used in paraphrasing and rephrasing. Some methods of deep learning such as seq-2-seq, transformers-based architecture (BERT) is used in summarization. This technique can produce more human-like summaries and handles paraphrasing also able to create more meaningful and concise summary. The issues faced are to maintain factual accuracy and produces incorrect information and also requirement of more complex models which can make computation expensive.

**Hybrid Based Text Summarization -** It combines elements of both extractive and abstractive approaches to achieve better results which provides more accurate and readable summaries. The main issue can be complexity which increases due to a combination of different methods.

Text Summarization holds significant benefits such as:

1. **Time saving:** It allows users to quickly understand the content without investing a lot of time in reading the entire document.
2. **Information retrieval:** Summaries act as useful abstracts that help users decide whether to delve deeper into a full document.
3. **Data analysis:** Summarization is valuable for analysing large datasets and extract meaningful insights without reading the whole document.
4. **Information dissemination:** Important news or any updates are conveyed into a short script to a broader audience.

## 3. Literature Review

This section describes various strategies for keyword extraction, such as supervised or unsupervised algorithms, clustering-based algorithms, and graph-based algorithms to generate an effective summary.

Christian Hans et al. [1] have used the TF-IDF approach to extract keywords. By calculating each sentence's score, the sentence that consists of the most important keywords is chosen as a summary text. Yohei [2] also uses the TF-IDF technique on biased terms such as nouns, words in the headers, and the position of the line in the document to extract keywords from the document and generate summary.

TextSum [3] has used the term frequency (TF) and inverted sentence frequency (ISF) approaches for keyword extraction. It uses the statistical approach, keyword extraction methods, and unsupervised learning techniques. It provides an average accuracy of around 83.03% as an effective summary approach.

Jo, T. et al. [6] provided the idea of extracting keywords using backpropagation neural networks. It considers key features such as title, first and last sentence words, TF, IDF, and ITF that are provided to the neural network model. This concludes that the threshold of around 0.51 and 0.49 can approximately classify keywords (that provide relevant information) from non-keywords (that are not relevant enough). While Sinha A. et al. [5] have also used word embedding methods to convert the text into a numerical value using a fast-text technique that provides a vector representation of words generating a summary of limited length.

Kaikhah, K. [4] created a set of seven features that are used to train the neural network model. The 3-step process follows pre-processing (stemming, stop-word removal) with a set of features such as the number of words in the title, paragraph title, location of paragraph in a document, first and last sentences, etc. Secondly, using a three-layered feedforward network, it trains the data on given features. After training, pruning is done, which means eliminating the few features, and feature fusion, that is clusters of hidden units are generated. Lastly, relevant sentences are extracted, providing 96% accuracy in summary generation.

Other than neural network clustering-based methods like k-means used by Haider, M. [11], and Khan, R. [12], the former uses word2vec, which is a word embedding technique to transform a word into a vector representation, while the later uses TF-IDF technique to extract key features from the whole document.

While graph-based methods such as the TextRank algorithm is an unsupervised graph-based text summarizing method that employs an extractive approach. Search engines like Google employ the PageRank algorithm to determine the ranking of online sites. The PageRank algorithm is the foundation of TextRank. The paper by Gunasundari S. et al. [14] discusses the TextRank approach with cosine similarity to avoid duplicate copies of sentences in the summary.

Other built-in modules, such as SPACY and NLP (natural language processing), are used to build extractive-based summary models, discussed in papers [15] and [16].

## 4. Research Methodology

This section discuss about algorithms, dataset collection method and low or process chart of text summarization.

### 4.1 Extractive Based Algorithms

All three algorithms TF-IDF, GloVe and TextRank are Extractive based approach to generate summary. The algorithms used follows no training phase to learn the data which can sometimes be a time-consuming process. Using scoring techniques it just scores each sentences and extract top N sentences as a summary which makes sense to the article.

#### 4.1.1 TF-IDF

TF-IDF ( Term Frequency – Inverse Document Frequency) is a numerical statistical method which describes the importance of a word in the document. The TF-IDF value increases proportionally to the number of times when a word appears in the document, but it is offset by the frequency of the word in the corpus, which helps to control the fact that some words are more common than others. The frequency term means the raw frequency of a term in a document. Moreover, the term regarding inverse document frequency is a measure of whether the term is common or rare across all documents which can be obtained by dividing the total number of documents by the number of documents containing the term by [2].

Formula to calculated TF-IDF of each term is,

(1)

(2)

(3)

The values of TF-IDF ranges from 0 to 1 with 10 digit-precision. After calculating the score of each words these are stored in a matrix in descending order. The importance of a sentence is calculated by sum of the scores of words. The sentences are then sorted in descending order taking only top N sentences with highest score as summary. The logic is “**sentences having a set words with high values, increases its chances to be in a summary”.**

#### 4.1.2 Word Embedding

Word Embedding in NLP is an important term that is used for representing words for text analysis in the form of real-valued vectors. It is an advancement in NLP that has improved the ability of computers to understand text-based content in a better way. It is considered one of the most popular approach in deep learning to solve challenges in Natural Language Processing.

In this approach, words and documents are represented in the form of numeric vectors allowing similar words to have similar vector representations. The extracted features are fed into machine learning model so as to work with text data and preserve the semantic and syntactic information. This information once received in its converted form is used by NLP algorithms that easily digest these learned representations and process textual information.

Various methods like Word2Vec, FastText and GloVe are used which saves training time provides a pre-trained model for performing many applications and one such application is text summarization.

#### 54.1.3 GloVe

GloVe (Global Vectors for Word Representation) is primarily employed for generating word embeddings, which are dense vector representation of words. These word embeddings capture semantic relationships between words based on their co-occurrence patterns in a large text corpus.

Glove helps in understanding the semantic relationship between words which helps in identifying important words in a text. It is a crucial step in text summarization. It helps to extract relevant keywords based on the context of the entire document.

To achieve text summarization using GloVe a 3-Step approach is done.

1. Load the pre-trained GloVe model. Each word in the embeddings is associated with a dense vector representation which is stored in a dictionary.
2. For each sentence in the article a sentence embedding is created. To calculate sentence embeddings for each sentence, a sentence is split into words. It takes the vector representation of each words. If that word is not found a zero vector is used as a placeholder. The resulting sentence embedding consist the mean value of all the words as a dense vector representation.
3. A cosine similarity matrix is created with take sentence embeddings as input. A row is represented as a sentences and column consists value of words. Each row value is summed up and a final score is generated for each row. All of the row values are sorted in the descending order and top N sentences is chosen for the summary. A sentence with a highest value is consider to be a part of the summary.

#### 4.1.4 TextRank

TextRank is an extractive text summarization and keyword extraction algorithm that is widely used in Natural Language Processing and Information Retrieval. It was introduced by Mihalcea and Tarau in 2004 and is inspired by Google’s PageRank algorithms, which ranks web pages in search engine results. TextRank is mainly used for automatically summarizing a larger text or identifying the most important keywords or phrases in the documents.

The graph-based algorithm is proven to be effective for extractive text summarization. For generating summaries using TextRank algorithm a graph is created where nodes represent sentences or words, and edges represent the relationships between them. The graph is used to calculate the scores of nodes and finding nodes with high values.

Steps to generate summaries using TextRank algorithm

1. Creating a vocabulary which consist set of unique words from the article.
2. Creating a co-occurrence matrix where each row represents a sentence and column represents a word in the vocabulary. The matrix is populated with word counts, meaning frequency of word occurring in each sentence.
3. Cosine Similarity is used to calculate cosine similarity between sentences based on the co-occurrence matrix.

(4)

1. TextRank similarity is applied on sentences to rank them based on their similarity. It runs for k iterations to update the sentence score.

Score(Sj) = ( (5)

1. Finally, the scores generated from the algorithm sorted in descending order considering top N highest score sentences to generate summary.

### 4.2 Data Collection

BBC news dataset is used which consists title and summaries and articles. It consists 5 domain which are Tech, Entertainment, Sport, Politics, Business.

|  |  |
| --- | --- |
| Domain | No. of Articles and summaries |
| Business | 510 |
| Tech | 401 |
| Sport | 511 |
| Entertainment | 386 |
| Politics | 417 |

Table 1: BBC news Dataset having number of articles in each domain

### 4.3 Process flow diagram

**Pre-processing**

**Feature Extraction (POS tags)**

**Algorithm**

**Sentence Scoring**

**Summary**

Figure 1: The process flow diagram of text summary generation.

#### 4.3.1 Pre-processing

The article is tokenized into words and those words are either removed or consider as keyword for summary generation. This step uses stop-words from NLTK package to remove less relevant words such as (can, is, at, the, and). Also using contraction module which consists full form of clitic words that is used for converting the words like can’t too cannot after which stop-words is applied for better summary. These provide a better word that are used for generating summaries.

#### 4.3.2 Feature Extraction

One more approach can be added is to only consider pronouns and nouns (actions words) increasing the performance of summary on different algorithms as shown in figure 3below. Few algorithms performs better without POS tags too. In results section it provides those algorithms which works better with or without POS tags.

#### 4.3.3 Algorithms

All three algorithms are extractive based approach. These takes input generated after pre-processing and feature extraction and compute probabilities of each term. These probabilities or score of each term tells whether or not a word can be a part of the summary.

#### 4.3.4 Sentence Scoring

In this step, the top most N sentences are considered as a summary. To get important sentences probabilities or score of words from previous step is used. A sentence score is computed by aggregating all values of words. Finally, a top N sentence is chosen in descending order considering higher values.

#### 4.3.5 Summary

Finally, the last step produce string of important sentences as a summary. A algorithms whose summary has highest F1 score than others is chosen to be a better summary.

## 5. Experimental Setup

The project Development is done using

Language :- Python

Coding software :- Python IDLE

Modules :- CustomTkinter and Tkinter for GUI of the application.

## 6. Results

In this section we will see the behaviour of different algorithms in terms of selecting an important sentence as their summaries. The summary generated by these algorithms are matched against summaries available in the dataset for each article. For evaluation of the models, F1 Score, Precision and Recall procedure is used.

F1 score is an evaluation metric that measures a model's accuracy. It combines the precision and recall scores of a model. The accuracy metric computes how many times a model made a correct prediction across the entire dataset.

All of the graphs below shows the comparison of TF-IDF, GloVe Embedding and TextRank. To check performance of each method only 10 articles of 5 domains in BBC news dataset which are entertainment, sports, tech, business, politics is used. Using F1 score metric all the below graphs shows the best method to choose for generating summaries of various articles. These F1 scores are calculate using average of all F1 scores computed from 5 domain.

Figure 2 shows the F1 score of each method. It tells that GloVe works better than TF-IDF and TextRank in generating summaries for articles.

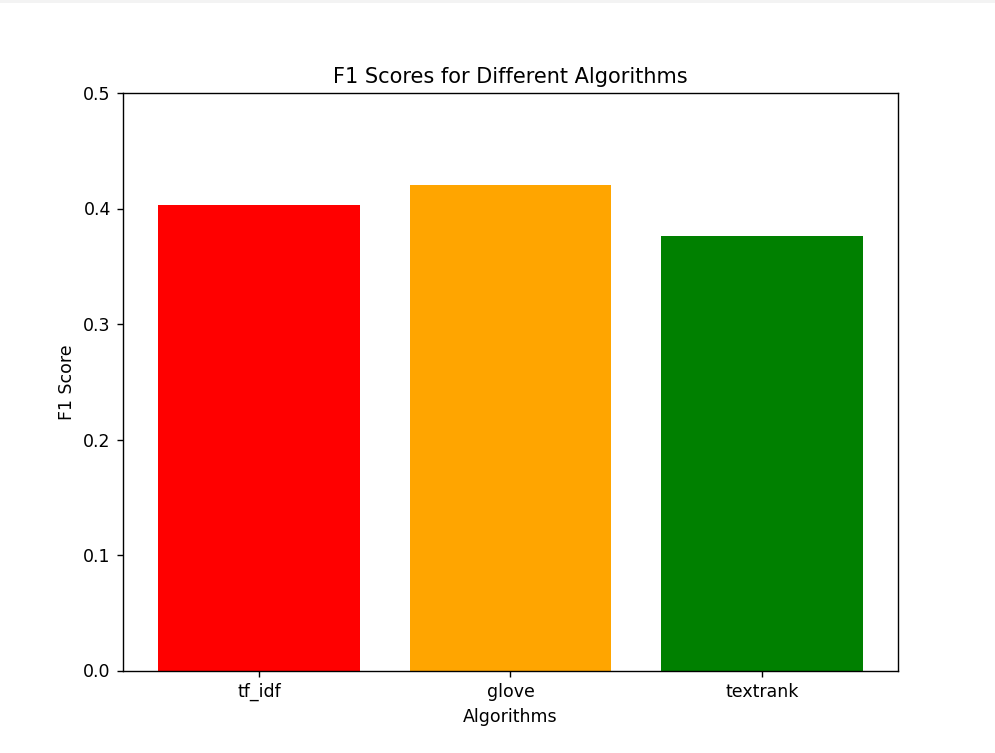


Figure 2: Average F1 Score of different algorithms.

In figure 3, the average F1 score is calculated for each method on each domain. The graph concludes that GloVe performs best on every domain except Tech.TF-IDF works better on Tech domain.

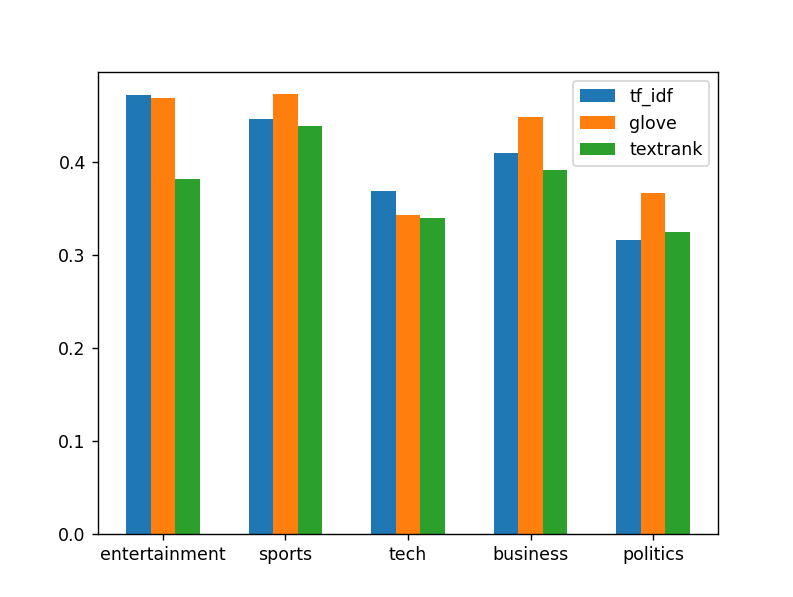


Figure 3: Comparison of methods on 5 domains.

Use of POS tags ( parts of speech tags ) as a feature extraction is done to get better results on summarization of articles. Since, it is stated that **sentences containing proper nouns and pronouns have greater chances to be included** in the summary by [1], all of the methods are evaluated using parts of speech tags. Figure 4 concludes that TextRank achieve better F1 score than TF-IDF and GloVe.

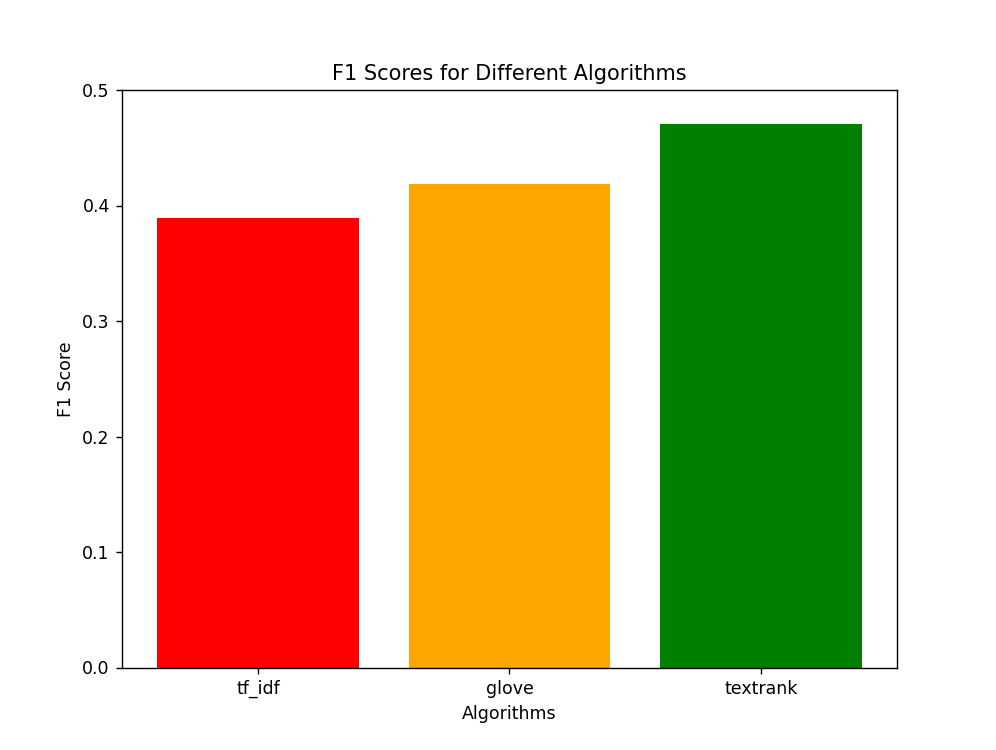


Figure 4: F1 Score of algorithms using POS tags.

Based on figure 5 results it is concluded that if we apply POS tags with TextRank it works better on every domain generating better summaries than other two methods.

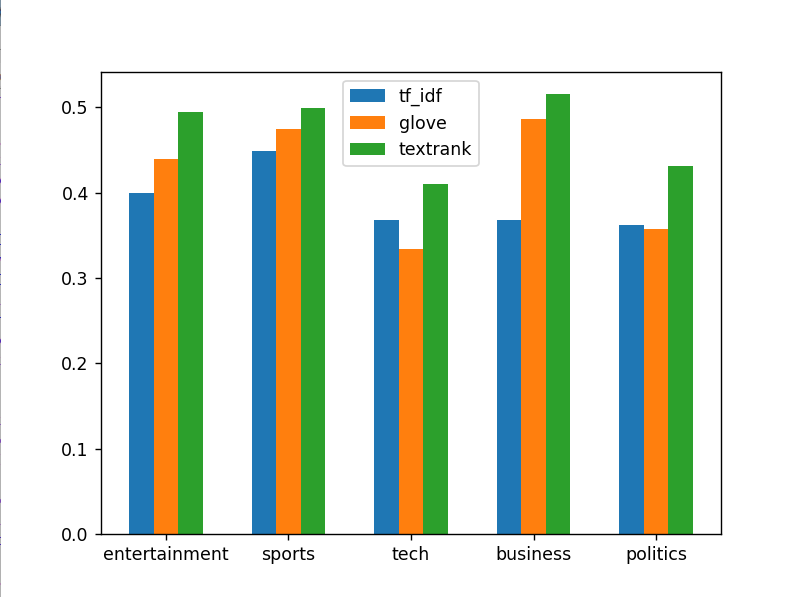


Figure 5: Comparison of methods on 5 domains using POS tags in pre-processing.

Using F1 score of every 5 domain of figure 3 and figure 5 an average score of each algorithm is calculated based on with and without POS tags values. The results are shown in figure 6 where TextRank accuracy is improved while Glove remains somewhat same. Also TF-IDF provides a slight improvement in its score. This graph helps to conclude that TextRank generated summaries can be improve with POS tags to achieve better performance while GloVe and TF-IDF algorithm can be kept same.

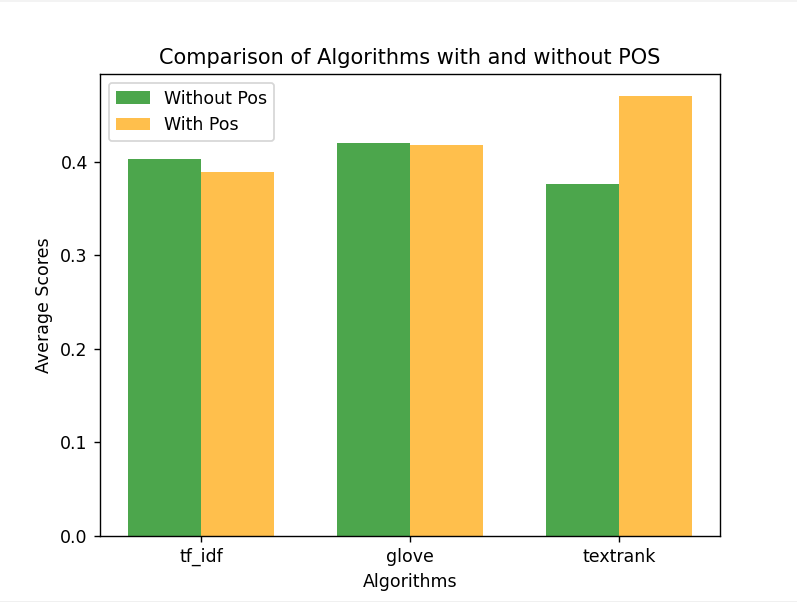


Figure 6: Comparison of algorithms from their previous performance with and without POS tags.

### 6.1 Code

#### 6.1.1 TF\_IDF Method

import numpy as np

import pandas as pd

from operator import itemgetter

import pre\_processing as pp

import nltk

from nltk.tokenize import sent\_tokenize, word\_tokenize

import matplotlib.pyplot as plt

class TF\_IDF :

def \_\_init\_\_(self,all\_documents):

self.all\_documents = all\_documents

self.TermFreq = {}

self.InvertedDocFreq = {}

def TermFrequency(self):

for document in self.all\_documents:

words = document.split()

for word in words:

tf = round(words.count(word)/len(words),4)

self.TermFreq[word] = tf

return self.TermFreq

def InvertedDocumentFrequency(self):

for document in self.all\_documents:

for term in document.split():

num\_docs\_with\_term = sum(1 for doc in self.all\_documents if term in doc)

idf = np.log(len(self.all\_documents) / num\_docs\_with\_term)

self.InvertedDocFreq[term] = idf

return self.InvertedDocFreq

def result\_tf\_idf(self):

get\_tf\_idf = {}

terms\_all\_docs = list(set(w for sent in self.all\_documents for w in sent.split()))

number\_of\_documents = len(self.all\_documents)

# get tf\_idf values of all terms

for term in terms\_all\_docs:

get\_tf\_idf[term] = round(self.TermFreq[term] \* self.InvertedDocFreq[term],4)

# create a dataframe for tf-idf matrix

tf\_idf\_table = pd.DataFrame(columns = terms\_all\_docs,index = np.arange(number\_of\_documents))

# add values in the dataframe

for document in self.all\_documents:

for term in tf\_idf\_table:

if (term in document):

tf\_idf\_table.loc[self.all\_documents.index(document)][term] = get\_tf\_idf[term]

else:

tf\_idf\_table.loc[self.all\_documents.index(document)][term] = 0

return tf\_idf\_table

def sentence\_scoring(article,tf\_idf\_table):

sentences = sent\_tokenize(article)

filtered\_sentences = []

for sentence in sentences:

res = " ".join(pp.preprocessing(sentence))

filtered\_sentences.append(res)

# calculate the scores for each sentences of the text

sentences\_scores = {} # Dictionary to store sentence scores

for sentence\_idx in range(len(filtered\_sentences)):

words = word\_tokenize(filtered\_sentences[sentence\_idx]) # sentence word tokens

# Calculate the sentence score using TF-IDF values

sentence\_score = 0

for word in words:

if word in tf\_idf\_table.columns: # Check if the word exists in TF-IDF columns

sentence\_score += tf\_idf\_table.loc[0][word]

sentences\_scores[sentence\_idx] = round(sentence\_score,4)

# selecting the top sentences as a summary

#sentence\_in\_summary = int((/100)\*len(sentences))

res = dict(sorted(sentences\_scores.items(), key=itemgetter(1), reverse=True)[:5]) # get top 5 keys having high values

res = dict(sorted(res.items()))# sorting the keys in sequence

summary = ''

for keys in res:

summary += ' '+ sentences[keys]

return summary

def get\_input\_text(article):

clean\_text = pp.preprocessing(article)

# Calculate TF-IDF

t1 = TF\_IDF(clean\_text)

term\_freq = t1.TermFrequency()

inverted\_doc\_freq = t1.InvertedDocumentFrequency()

tf\_idf = t1.result\_tf\_idf()

summary = sentence\_scoring(article,tf\_idf)

return summary

#### 6.1.2 GloVe Method

from sklearn.metrics.pairwise import cosine\_similarity

import numpy as np

import pandas as pd

import nltk

from nltk.tokenize import sent\_tokenize, word\_tokenize

import pre\_processing as pp

from nltk import pos\_tag

# Load the pre-trained GloVe embeddings

def load\_glove\_embeddings(file\_path):

word\_embeddings = {}

with open(file\_path, encoding='utf-8') as f:

for line in f:

values = line.split()

word = values[0]

embedding = np.asarray(values[1:], dtype='float32')

word\_embeddings[word] = embedding

return word\_embeddings

# Load pre-trained GloVe embeddings (adjust path accordingly)

glove\_embeddings = load\_glove\_embeddings('glove6/glove.6B.300d.txt')

# Calculate sentence embedding using average of word embeddings

def calculate\_sentence\_embedding(sentence, word\_embeddings):

words = sentence.split()

sentence\_embedding = np.mean([word\_embeddings.get(word, np.zeros(word\_embeddings['a'].shape)) for word in words], axis=0)

return sentence\_embedding

def get\_input\_text2(article):

first\_sent\_idx = 0

clean\_text = pp.preprocessing(article)

sentence\_embeddings = []

for sentence in clean\_text:

sent\_embedding = calculate\_sentence\_embedding(sentence,glove\_embeddings)

sentence\_embeddings.append(sent\_embedding)

similarity\_matrix = cosine\_similarity(sentence\_embeddings)

num\_summary\_sentences = 5

summary\_indices = np.argsort(np.sum(similarity\_matrix,axis = 1))[-num\_summary\_sentences:]

summary\_indices = sorted(summary\_indices)

original\_text = sent\_tokenize(article)

summary\_sentences = [ original\_text[idx] for idx in summary\_indices]

summary\_res = "\n".join(summary\_sentences)

return summary\_res

#### 6.1.3 TextRank Method

import numpy as np

import pandas as pd

import re

import nltk

from nltk.tokenize import sent\_tokenize, word\_tokenize

from nltk.corpus import stopwords

import warnings

import contractions

import pre\_processing as pp

# Preprocess the text

stop\_words = set(stopwords.words('english'))

# Create a graph using the TextRank algorithm

def create\_graph(sentences):

words\_list = [pp.pos\_tagging(" ".join(pp.preprocessing(sentence))) for sentence in sentences]

words = set(word for words in words\_list for word in words)

word2id = {word: i for i, word in enumerate(words)}

id2word = {i: word for word, i in word2id.items()}

# Create a matrix where each row represents a sentence and each column represents a word

matrix = np.zeros((len(sentences), len(words)))

for i, words in enumerate(words\_list):

for word in words:

matrix[i][word2id[word]] += 1

# Calculate sentence similarity using cosine similarity

similarity\_matrix = np.dot(matrix, matrix.T)

norm\_matrix = np.outer(np.linalg.norm(matrix, axis=1), np.linalg.norm(matrix, axis=1))

similarity\_matrix = similarity\_matrix / norm\_matrix

# Apply the PageRank algorithm to rank sentences

damping\_factor = 0.85

scores = np.ones(len(sentences))

for \_ in range(100): # Number of iterations

scores = (1 - damping\_factor) + damping\_factor \* np.dot(similarity\_matrix, scores)

return scores

# Get the top N sentences as the summary

def get\_summary(sentences, scores, top\_n = 5):

ranked\_sentences = [(sentence, score) for sentence, score in zip(sentences, scores)]

ranked\_sentences.sort(key=lambda x: x[1], reverse=True)

summary\_sentences = ranked\_sentences[:top\_n]

summary = [sentence for sentence, \_ in summary\_sentences]

return " ".join(summary)

def get\_input\_text3(article):

sentences = nltk.sent\_tokenize(article)

# Generate the summary

scores = create\_graph(sentences)

predict\_summary = get\_summary(sentences, scores)

return predict\_summary

#### 6.1.4 Main file

import customtkinter

from customtkinter import StringVar, IntVar

from PIL import Image,ImageTk

from tkinter import filedialog

import tkinter

from CTkMessagebox import CTkMessagebox

from nltk import sent\_tokenize

import Method\_1\_TF\_IDF as tf\_idf

import Method\_2\_Glove as glove

import Method\_3\_TextRank as textrank

import pre\_processing as pp

import model\_evaluation as check\_acc

import os

title = ''

article = ''

summary = ''

category = ''

article\_filename = ''

res\_summry = ''

customtkinter.set\_appearance\_mode("Light") # Modes: "System" (standard), "Dark", "Light"

customtkinter.set\_default\_color\_theme("green") # Themes: "blue" (standard), "green", "dark-blue"

master = customtkinter.CTk() # customtkinter

master.geometry("1440x780")

master.title("Text Summarization")

master.resizable(False,False)

# Input File frame

input\_text = customtkinter.CTkTextbox(master,font=('arial',15),width = 650 ,height = 400)

input\_text.grid(row=7,column=7,columnspan=3,padx=48,pady=30)

# Input File frame

output\_summary = customtkinter.CTkTextbox(master,font=('arial',15),width = 650 ,height = 400)

output\_summary.grid(row=7,column=10)

score\_btn = customtkinter.CTkButton(master, text = '00.0%',font = ('arial',13),width = 35,height = 45,corner\_radius = 22,bg\_color = "#F8F8F8")

score\_btn.place(relx=0.56, rely=0.51, anchor=tkinter.CENTER)

def download\_summary():

res\_summary = output\_summary.get("0.0", "end")

filename\_path = "D:\\Research Papers\\TextSummary\\BBC News Summary\\GeneratedSummaries" + "\\" + category + "\\" + article\_filename

file = open(filename\_path,"w+")

file.write(res\_summary)

file.close()

img = tkinter.PhotoImage(file="download.png")

download\_btn = customtkinter.CTkButton(master, text = "", image = img,width = 4,height = 4,corner\_radius = 5,bg\_color = "#F8F8F8",fg\_color = "#F8F8F8",

hover\_color = "#F8F8F8",command = download\_summary)

download\_btn.place(relx=0.94, rely=0.51, anchor=tkinter.CENTER)

# Function to open a file and populate text\_box\_A

def open\_file():

global category

global article\_filename

file\_path = filedialog.askopenfilename(filetypes=[("Text files", "\*.txt")])

summary\_path = file\_path.split('/')

category = summary\_path[-2].strip()

article\_filename = summary\_path[-1].strip()

summary\_path[4] = 'Summaries'

summary\_path = "/".join(summary\_path)

global title

global article

global summary

if file\_path:

with open(file\_path, "r") as file:

title = next(file) # title of the doc

article = file.read()

input\_text.delete("1.0", customtkinter.END)

input\_text.insert(customtkinter.END, article)

if summary\_path:

with open(summary\_path,"r") as file2:

summary = file2.read()

# GET SUMMARIES OF HIGEST F1 SOCRE

def get\_summaries():

tf\_idf\_summary = tf\_idf.get\_input\_text(article)

tf\_idf\_acc = check\_acc.calculate\_f1\_score(tf\_idf\_summary,summary)

glove\_summary = glove.get\_input\_text2(article)

glove\_acc = check\_acc.calculate\_f1\_score(glove\_summary,summary)

textrank\_summary = textrank.get\_input\_text3(article)

textrank\_acc = check\_acc.calculate\_f1\_score(textrank\_summary,summary)

if((tf\_idf\_acc > glove\_acc ) and (tf\_idf\_acc > textrank\_acc)):

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,tf\_idf\_summary)

score\_btn.configure(text = str(tf\_idf\_acc)+"%")

elif((glove\_acc > tf\_idf\_acc) and (glove\_acc > textrank\_acc)):

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,glove\_summary)

score\_btn.configure(text = str(glove\_acc)+"%")

elif((textrank\_acc > tf\_idf\_acc ) and (textrank\_acc > glove\_acc )):

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,textrank\_summary)

score\_btn.configure(text = str(textrank\_acc)+"%")

else:

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,tf\_idf\_summary) # default summary method to show

score\_btn.configure(text = str(tf\_idf\_acc)+"%")

print("Accurcay tf\_idf\_result {} \n glove\_result {} \n textrank\_result {} ".format(tf\_idf\_acc,glove\_acc,textrank\_acc))

radio\_var = tkinter.IntVar(value=0)

# GET SUMMARIES BASED ON THE METHOD SELETEC

def Get\_Method():

if (radio\_var.get() == 1):

tf\_idf\_summary = tf\_idf.get\_input\_text(article)

tf\_idf\_acc = check\_acc.calculate\_f1\_score(tf\_idf\_summary,summary)

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,tf\_idf\_summary)

score\_btn.configure(text = str(tf\_idf\_acc)+"%")

elif(radio\_var.get() == 2):

glove\_summary = glove.get\_input\_text2(article)

glove\_acc = check\_acc.calculate\_f1\_score(glove\_summary,summary)

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,glove\_summary)

score\_btn.configure(text = str(glove\_acc)+"%")

elif(radio\_var.get() == 3):

textrank\_summary = textrank.get\_input\_text3(article)

textrank\_acc = check\_acc.calculate\_f1\_score(textrank\_summary,summary)

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,textrank\_summary)

score\_btn.configure(text = str(textrank\_acc)+"%")

else:

output\_summary.delete("1.0", customtkinter.END)

output\_summary.insert(customtkinter.END,"No summaries")

score = '0'

# Use CTkButton instead of tkinter Button

upload\_btn = customtkinter.CTkButton(master, text="Upload File",font = ('arial',15),width = 40,height = 40, command = open\_file)

upload\_btn.grid(row = 10, column = 8,pady = 10)

summary\_btn = customtkinter.CTkButton(master, text="Get Summary",font = ('arial',15),width = 40,height = 40,command = get\_summaries)

summary\_btn.grid(row = 10, column = 10)

label\_radio\_group = customtkinter.CTkLabel(master, text="Select Summary Method")

label\_radio\_group.place(relx=0.1, rely=0.7)

radio\_button\_1 = customtkinter.CTkRadioButton(master, variable=radio\_var, value = 1,text = "TF-IDF",command = Get\_Method)

radio\_button\_1.place(relx=0.3, rely=0.7)

radio\_button\_2 = customtkinter.CTkRadioButton(master, variable=radio\_var, value = 2,text = "Glove",command = Get\_Method)

radio\_button\_2.place(relx=0.4, rely=0.7)

radio\_button\_3 = customtkinter.CTkRadioButton(master, variable=radio\_var, value = 3,text = "TextRank",command = Get\_Method)

radio\_button\_3.place(relx=0.5, rely=0.7)

master.mainloop()

### 6.2 GUI Output

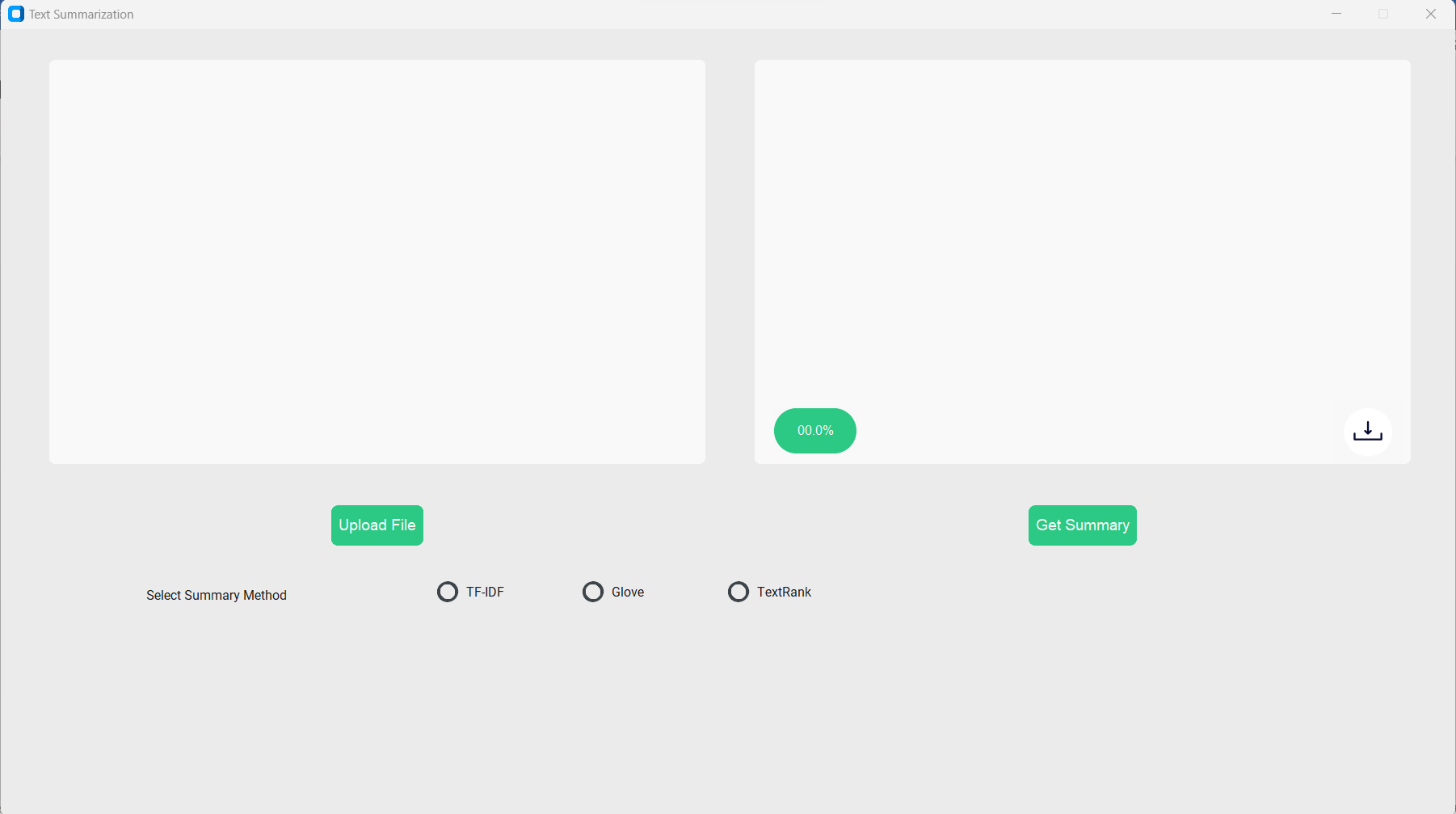


Figure 7: GUI of Text Summary

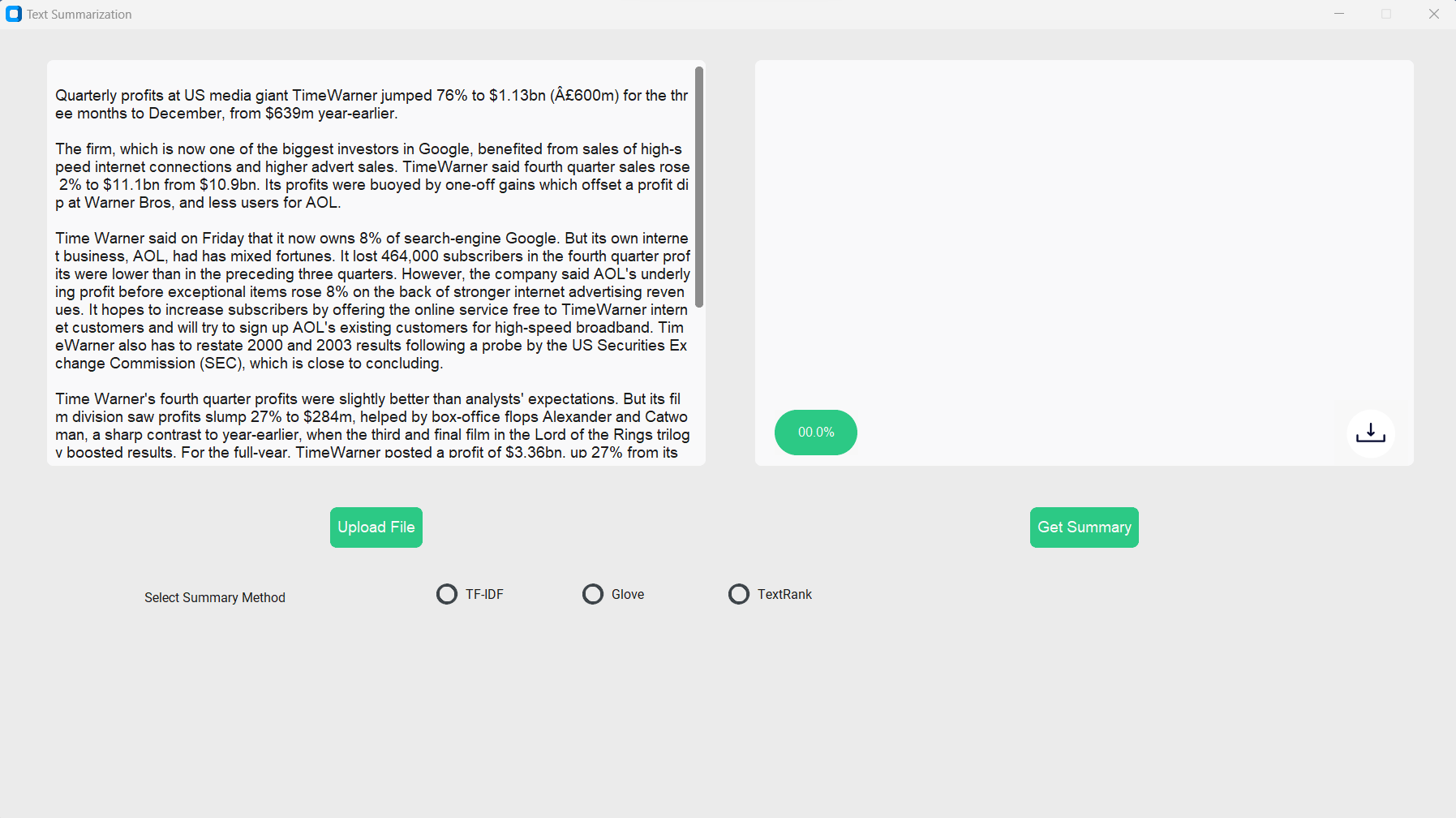


Figure 8: Upload an Article File

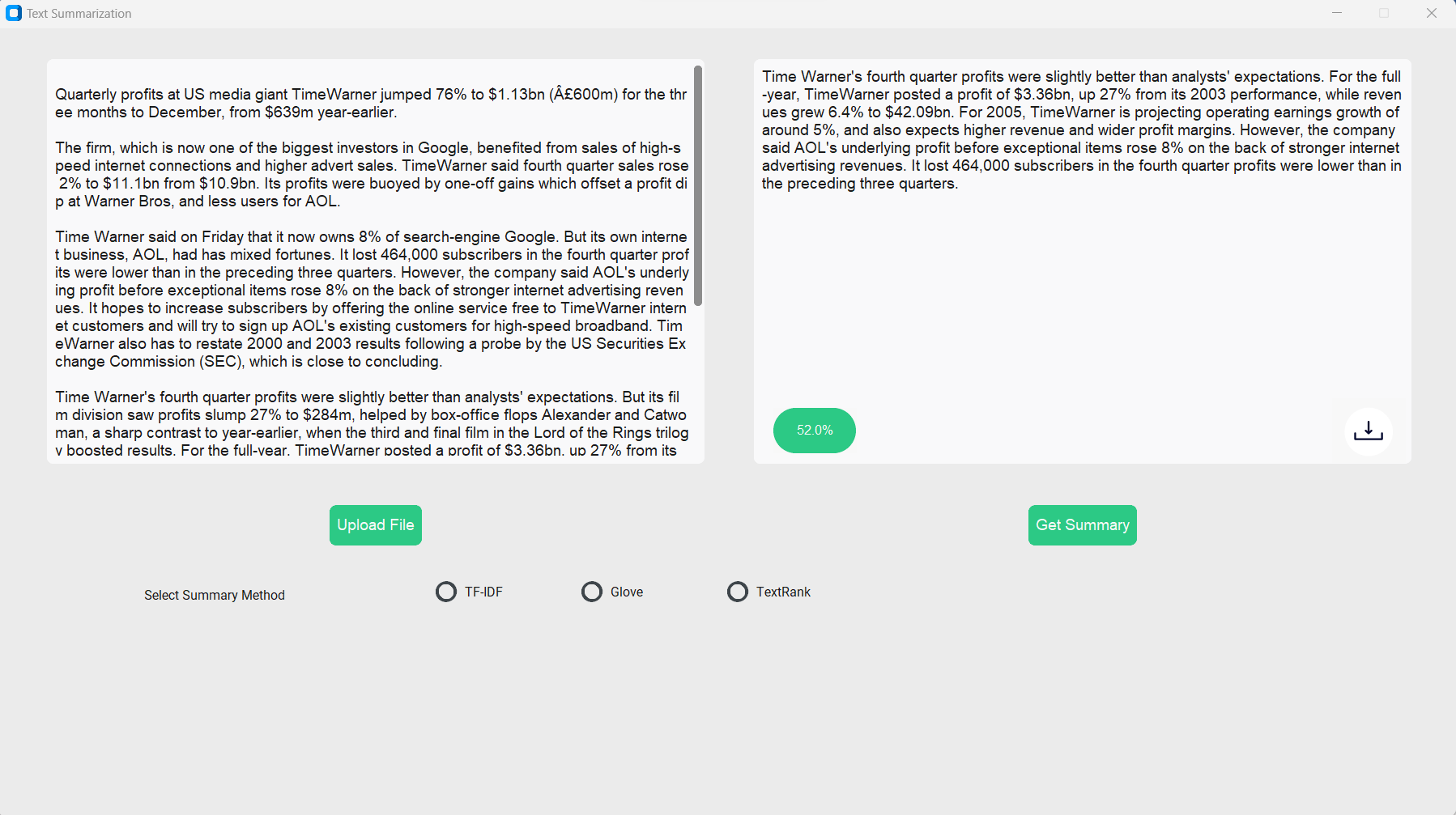


Figure 9: Get Summary of the Article

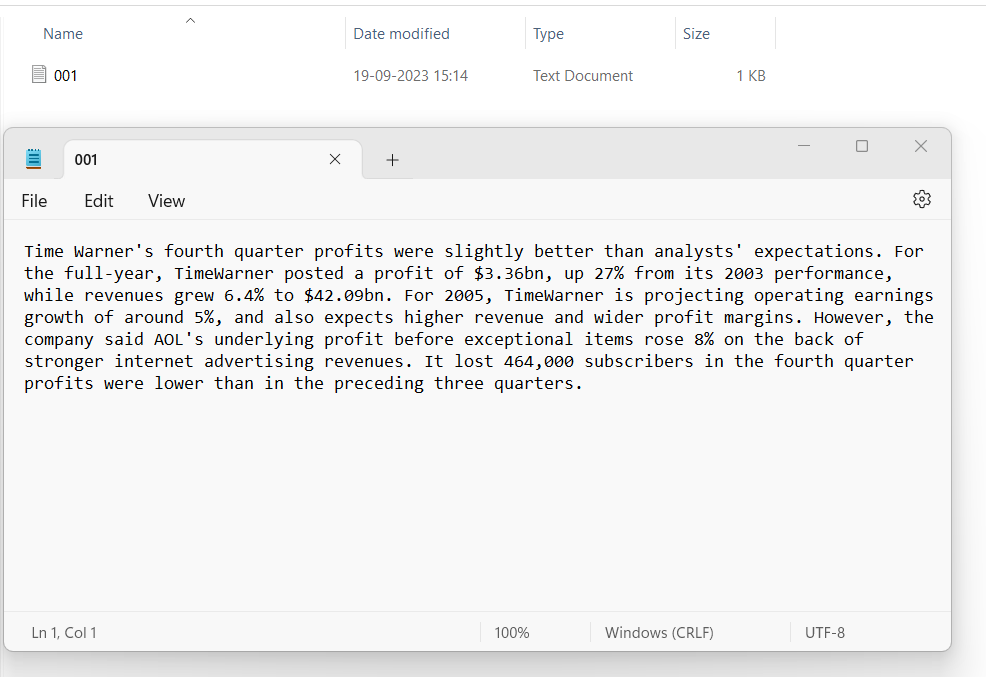


Figure 10: Download summary of the Article

### 6.3 Summary of articles for each domain

|  |  |
| --- | --- |
| **Table 1: Entertainment Article** | |
| Article | A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.  The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate. The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. It is the 17th year that the gallery has invited an artist to dress their Christmas tree. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.    The plain green Norway spruce is displayed in the gallery's foyer. Its light bulb adornments are dimmed, ordinary domestic ones joined together with string. The plates decorating the branches will be auctioned off for the children's charity ArtWorks. Wentworth worked as an assistant to sculptor Henry Moore in the late 1960s. His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades. Wentworth is also known for his photography of mundane, everyday subjects such as a cigarette packet jammed under the wonky leg of a table. |
| Original Summary | The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery. It is the 17th year that the gallery has invited an artist to dress their Christmas tree. The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate. His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades. |
| TF\_IDF Summary | A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery. The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate. The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. It is the 17th year that the gallery has invited an artist to dress their Christmas tree. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002. |
| GloVe Summary | A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery.  The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs.  It is the 17th year that the gallery has invited an artist to dress their Christmas tree.  Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002.  His reputation as a sculptor grew in the 1980s, while he has been one of the most influential teachers during the last two decades. |
| Text Rank Summary | A Christmas tree that can receive text messages has been unveiled at London's Tate Britain art gallery. The spruce has an antenna which can receive Bluetooth texts sent by visitors to the Tate. The messages will be "unwrapped" by sculptor Richard Wentworth, who is responsible for decorating the tree with broken plates and light bulbs. It is the 17th year that the gallery has invited an artist to dress their Christmas tree. Artists who have decorated the Tate tree in previous years include Tracey Emin in 2002. |

|  |  |
| --- | --- |
| **Table 2: Sport Article** | |
| Article | British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid.  The 25-year-old has already smashed the British record over 60m hurdles twice this season, setting a new mark of 7.96 seconds to win the AAAs title. "I am quite confident," said Claxton. "But I take each race as it comes. "As long as I keep up my training but not do too much, I think there is a chance of a medal." Claxton has won the national 60m hurdles title for the past three years but has struggled to translate her domestic success to the international stage. Now, the Scotland-born athlete owns the equal fifth-fastest time in the world this year. And at last week's Birmingham Grand Prix, Claxton left European medal favourite Russian Irina Shevchenko trailing in sixth spot.  For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form. In previous seasons, the 25-year-old also contested the long jump but since moving from Colchester to London she has re-focused her attentions. Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. |
| Original Summary | For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form .Claxton has won the national 60m hurdles title for the past three years but has struggled to translate her domestic success to the international stage. British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. "I am quite confident," said Claxton. |
| TF\_IDF Summary | British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. "I am quite confident," said Claxton. And at last week's Birmingham Grand Prix, Claxton left European medal favourite Russian Irina Shevchenko trailing in sixth spot. For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form. Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. |
| GloVe Summary | British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid.  Claxton has won the national 60m hurdles title for the past three years but has struggled to translate her domestic success to the international stage.  Now, the Scotland-born athlete owns the equal fifth-fastest time in the world this year.  For the first time, Claxton has only been preparing for a campaign over the hurdles - which could explain her leap in form.  Claxton will see if her new training regime pays dividends at the European Indoors which take place on 5-6 March. |
| Text Rank Summary | British hurdler Sarah Claxton is confident she can win her first major medal at next month's European Indoor Championships in Madrid. The 25-year-old has already smashed the British record over 60m hurdles twice this season, setting a new mark of 7.96 seconds to win the AAAs title. "I am quite confident," said Claxton. "But I take each race as it comes. "As long as I keep up my training but not do too much, I think there is a chance of a medal." |

|  |  |
| --- | --- |
| **Table 3: Tech Article** | |
| Article | US gamers will be able to buy Sony's PlayStation Portable from 24 March, but there is no news of a Europe debut.  The handheld console will go on sale for $250 (Â£132) and the first million sold will come with Spider-Man 2 on UMD, the disc format for the machine. Sony has billed the machine as the Walkman of the 21st Century and has sold more than 800,000 units in Japan. The console (12cm by 7.4cm) will play games, movies and music and also offers support for wireless gaming. Sony is entering a market which has been dominated by Nintendo for many years.  It launched its DS handheld in Japan and the US last year and has sold 2.8 million units. Sony has said it wanted to launch the PSP in Europe at roughly the same time as the US, but gamers will now fear that the launch has been put back. Nintendo has said it will release the DS in Europe from 11 March. "It has gaming at its core, but it's not a gaming device. It's an entertainment device," said Kaz Hirai, president of Sony Computer Entertainment America. |
| Original Summary | Sony has billed the machine as the Walkman of the 21st Century and has sold more than 800,000 units in Japan.Sony has said it wanted to launch the PSP in Europe at roughly the same time as the US, but gamers will now fear that the launch has been put back.Nintendo has said it will release the DS in Europe from 11 March.It launched its DS handheld in Japan and the US last year and has sold 2.8 million units. |
| TF\_IDF Summary | US gamers will be able to buy Sony's PlayStation Portable from 24 March, but there is no news of a Europe debut. The console (12cm by 7.4cm) will play games, movies and music and also offers support for wireless gaming. Sony has said it wanted to launch the PSP in Europe at roughly the same time as the US, but gamers will now fear that the launch has been put back. Nintendo has said it will release the DS in Europe from 11 March. "It has gaming at its core, but it's not a gaming device. |
| GloVe Summary | US gamers will be able to buy Sony's PlayStation Portable from 24 March, but there is no news of a Europe debut.  The handheld console will go on sale for $250 (Â£132) and the first million sold will come with Spider-Man 2 on UMD, the disc format for the machine.  The console (12cm by 7.4cm) will play games, movies and music and also offers support for wireless gaming.  It launched its DS handheld in Japan and the US last year and has sold 2.8 million units.  Sony has said it wanted to launch the PSP in Europe at roughly the same time as the US, but gamers will now fear that the launch has been put back. |
| Text Rank Summary | It launched its DS handheld in Japan and the US last year and has sold 2.8 million units. Nintendo has said it will release the DS in Europe from 11 March. Sony has billed the machine as the Walkman of the 21st Century and has sold more than 800,000 units in Japan. Sony has said it wanted to launch the PSP in Europe at roughly the same time as the US, but gamers will now fear that the launch has been put back. The handheld console will go on sale for $250 (Â£132) and the first million sold will come with Spider-Man 2 on UMD, the disc format for the machine. |

|  |  |
| --- | --- |
| **Table 4: Business Article** | |
| Article | India's rupee has hit a five-year high after Standard & Poor's (S&P) raised the country's foreign currency rating.  The rupee climbed to 43.305 per US dollar on Thursday, up from a close of 43.41. The currency has gained almost 1% in the past three sessions. S&P, which rates borrowers' creditworthiness, lifted India's rating by one notch to 'BB+'. With Indian assets now seen as less of a gamble, more cash is expected to flow into its markets, buoying the rupee.  "The upgrade is positive and basically people will use it as an excuse to come back to India," said Bhanu Baweja, a strategist at UBS. "Money has moved out from India in the first two or three weeks of January into other markets like Korea and Thailand and this upgrade should lead to a reversal." India's foreign currency rating is now one notch below investment grade, which starts at 'BBB-'. The increase has put it on the same level as Romania, Egypt and El Salvador, and one level below Russia. |
| Original Summary | India's rupee has hit a five-year high after Standard & Poor's (S&P) raised the country's foreign currency rating.India's foreign currency rating is now one notch below investment grade, which starts at 'BBB-'.S&P, which rates borrowers' creditworthiness, lifted India's rating by one notch to 'BB+'.The currency has gained almost 1% in the past three sessions. |
| TF\_IDF Summary | India's rupee has hit a five-year high after Standard & Poor's (S&P) raised the country's foreign currency rating. The rupee climbed to 43.305 per US dollar on Thursday, up from a close of 43.41. The currency has gained almost 1% in the past three sessions. S&P, which rates borrowers' creditworthiness, lifted India's rating by one notch to 'BB+'. India's foreign currency rating is now one notch below investment grade, which starts at 'BBB-'. |
| GloVe Summary | The rupee climbed to 43.305 per US dollar on Thursday, up from a close of 43.41.  The currency has gained almost 1% in the past three sessions.  S&P, which rates borrowers' creditworthiness, lifted India's rating by one notch to 'BB+'.  "Money has moved out from India in the first two or three weeks of January into other markets like Korea and Thailand and this upgrade should lead to a reversal."  India's foreign currency rating is now one notch below investment grade, which starts at 'BBB-'. |
| Text Rank Summary | India's rupee has hit a five-year high after Standard & Poor's (S&P) raised the country's foreign currency rating. India's foreign currency rating is now one notch below investment grade, which starts at 'BBB-'. S&P, which rates borrowers' creditworthiness, lifted India's rating by one notch to 'BB+'. The currency has gained almost 1% in the past three sessions. The rupee climbed to 43.305 per US dollar on Thursday, up from a close of 43.41. |

|  |  |
| --- | --- |
| **Table 5: Politics Article** | |
| Article | Tsunami debt deal to be announced  Chancellor Gordon Brown has said he hopes to announce a deal to suspend debt interest repayments by tsunami-hit nations later on Friday.  The agreement by the G8 group of wealthy nations would save affected countries £3bn pounds a year, he said. The deal is thought to have been hammered out on Thursday night after Japan, one of the biggest creditor nations, finally signed up to it. Mr Brown first proposed the idea earlier this week.  G8 ministers are also believed to have agreed to instruct the World Bank and the International Monetary Fund to complete a country by country analysis of the reconstruction problems faced by all states hit by the disaster. Mr Brown has been locked in talks with finance ministers of the G8, which Britain now chairs. Germany also proposed a freeze and Canada has begun its own moratorium. The expected deal comes as Foreign Secretary Jack Straw said the number of Britons dead or missing in the disaster have reached 440. |
| Original Summary | Mr Brown has been locked in talks with finance ministers of the G8, which Britain now chairs.Chancellor Gordon Brown has said he hopes to announce a deal to suspend debt interest repayments by tsunami-hit nations later on Friday.The agreement by the G8 group of wealthy nations would save affected countries £3bn pounds a year, he said.Mr Brown first proposed the idea earlier this week. |
| TF\_IDF Summary | Chancellor Gordon Brown has said he hopes to announce a deal to suspend debt interest repayments by tsunami-hit nations later on Friday. The agreement by the G8 group of wealthy nations would save affected countries Â£3bn pounds a year, he said. The deal is thought to have been hammered out on Thursday night after Japan, one of the biggest creditor nations, finally signed up to it. Mr Brown first proposed the idea earlier this week. The expected deal comes as Foreign Secretary Jack Straw said the number of Britons dead or missing in the disaster have reached 440. |
| GloVe Summary | Chancellor Gordon Brown has said he hopes to announce a deal to suspend debt interest repayments by tsunami-hit nations later on Friday.  The agreement by the G8 group of wealthy nations would save affected countries Â£3bn pounds a year, he said. The deal is thought to have been hammered out on Thursday night after Japan, one of the biggest creditor nations, finally signed up to it. Mr Brown first proposed the idea earlier this week. The expected deal comes as Foreign Secretary Jack Straw said the number of Britons dead or missing in the disaster have reached 440. |
| Text Rank Summary | Chancellor Gordon Brown has said he hopes to announce a deal to suspend debt interest repayments by tsunami-hit nations later on Friday. G8 ministers are also believed to have agreed to instruct the World Bank and the International Monetary Fund to complete a country by country analysis of the reconstruction problems faced by all states hit by the disaster. The agreement by the G8 group of wealthy nations would save affected countries Â£3bn pounds a year, he said. The deal is thought to have been hammered out on Thursday night after Japan, one of the biggest creditor nations, finally signed up to it. The expected deal comes as Foreign Secretary Jack Straw said the number of Britons dead or missing in the disaster have reached 440. |

## 7. Conclusion

This project report explores extractive based approaches to generate summaries where TextRank works best for sports and technology related domain articles. TD\_IDF provide better summaries across all article. Glove works netural better across all domains.

To improve these methods POS tags were used to extract more relevant features from the text. This improved the performance of TextRank method with F1 score of 0.47 which is better than TF-IDF and GloVe F1-Scores.

Finally, for the future scope to improve the extractive based summary generation. First, the feature extraction method shall be explored more such as title to make summary biased on the given title, first sentences and last sentences of the article to make sense of summary generated. This will improve the summary by only considering relevant sentences.

Second, to generate better summaries for huge articles and avoid fixed length of summaries. The number of sentences to be included in the summary was fixed which was top 5 sentence with highest score was chosen. This approach works better with shorter articles and may yield poor result with long articles.

## 8. References

1. Christian, H., Agus, M. P., & Suhartono, D. (2016). Single document automatic text summarization using term frequency-inverse document frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, *7*(4), 285-294.
2. Yohei, S. E. K. I. (2003). Sentence Extraction by tf/idf and Position Weighting from Newspaper. In *Proceedings of the Third NTCIR Workshop*.
3. Yong, S. P., Abidin, A. I., & Chen, Y. Y. (2006). A neural-based text summarization system. *WIT Transactions on Information and Communication Technologies*, *37*.
4. Kaikhah, K. (2004). Text summarization using neural networks.
5. Sinha, A., Yadav, A., & Gahlot, A. (2018). Extractive text summarization using neural networks. *arXiv preprint arXiv:1802.10137*.
6. Jo, T., Lee, M., & Gatton, T. M. (2006, November). Keyword extraction from documents using a neural network model. In *2006 International Conference on Hybrid Information Technology* (Vol. 2, pp. 194-197). IEEE.
7. Gupta, V., & Lehal, G. S. (2010). A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, *2*(3), 258-268.
8. Mridha, M. F., Lima, A. A., Nur, K., Das, S. C., Hasan, M., & Kabir, M. M. (2021). A survey of automatic text summarization: Progress, process and challenges. *IEEE Access*, *9*, 156043-156070.
9. Yadav, D., Desai, J., & Yadav, A. K. (2022). Automatic text summarization methods: A comprehensive review. *arXiv preprint arXiv:2204.01849*.
10. Haque, M. M., Pervin, S., & Begum, Z. (2013). Literature review of automatic single document text summarization using NLP. *International Journal of Innovation and Applied Studies*, *3*(3), 857-865.
11. Haider, M. M., Hossin, M. A., Mahi, H. R., & Arif, H. (2020, June). Automatic text summarization using gensim word2vec and k-means clustering algorithm. In *2020 IEEE Region 10 Symposium (TENSYMP)* (pp. 283-286). IEEE.
12. Khan, R., Qian, Y., & Naeem, S. (2019). Extractive based text summarization using k-means and tf-idf. *International Journal of Information Engineering and Electronic Business*, *10*(3), 33.
13. García-Hernández, R. A., & Ledeneva, Y. (2009, February). Word sequence models for single text summarization. In *2009 Second International Conferences on Advances in Computer-Human Interactions* (pp. 44-48). IEEE.
14. Gunasundari, S., Shylaja, M. J., Rajalaksmi, S., & Aarthi, M. K. IMPROVED DRIVEN TEXT SUMMARIZATION USING PAGERANKING ALGORITHM AND COSINE SIMILARITY.
15. Badgujar, A., Shaikh, A., Kamble, T., & Makhija, G. TEXT SUMMARIZATION USING NATURAL LANGUAGE PROCESSING.
16. JUGRAN, S., KUMAR, A., TYAGI, B. S., & ANAND, V. (2021, March). Extractive automatic text summarization using SpaCy in Python & NLP. In *2021 International conference on advance computing and innovative technologies in engineering (ICACITE)* (pp. 582-585). IEEE.

1. <https://nlp.stanford.edu/projects/glove/>