|  |  |  |
| --- | --- | --- |
|  | SRM INSTITUTE OF SCIENCE AND TECHNOLOGY  SCHOOL OF COMPUTING  DEPARTMENT OF DATA SCIENCE AND BUSINESS SYSTEMS  18CSC304J COMPILER DESIGN |  |
| **MINI PROJECT REPORT**  **TEXT SUMMARIZER** | | |
| **Name: Vardhit Saraogi**  **Register Number: RA1911027010028**  **Mail ID: vs9249@srmist.edu.in**  **Department: Data Science and Business Systems**  **Specialization: Big Data Analytics**  **Semester: 6th** | | |
| **Team Members’ Name : Shubhank Sargam, Harishankar PV**  **Registration Number : RA1911027010020, RA1911027010022** | | |

**Contents**

Abstract

Chapter 1: Introduction and Motivation [Purpose of the problem statement (societal benefit)]

Chapter 2: Proposed Method with System Architecture / Flow Diagram

Chapter 3: Modules Description

Chapter 4: Implementation requirements

Chapter 5: Output Screenshots

Conclusion

References

Appendix A – Source Code

Appendix B – GitHub Profile and Link for the Project

**Abstract**

The field of text summarization has been evolving along with advances in Natural Language Processing (NLP) and Computer Science but until now no known attempts have been made to create a summarizer for summarizing Icelandic texts. This project looks into the field of text summarization, giving an overview of different types and approaches towards automatically generating summaries.

**Chapter 1: Introduction and Motivation**

Summarizations play a crucial part in saving time. Books have gists written at the back-cover which entices the users. Sometimes the headings are enough to give away the meaning of the full paragraph it contains, movie reviews and trailers give a glimpse of what one might get to see in a movie.

The machines these days have started understanding human languages using Natural language processing. A text summarizer is an application that is helping us to shorten the given text to give a gist of what the text is trying to convey.

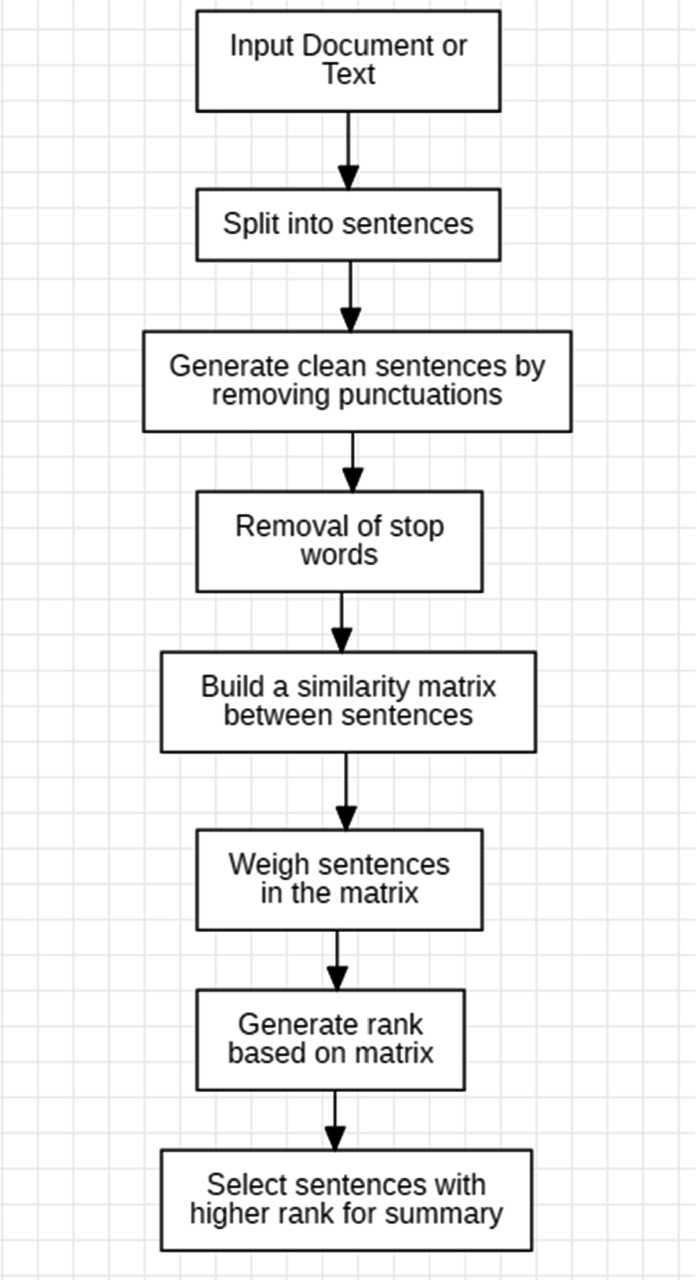
There are two types of summarizers, which are extractive and abstractive. The former may split the text into sentences and order them by their importance in the text. Pick higher rank sentences and add them to the summary. The latter analyses the texts and adds important topics to the newly created sentences which are then added to the summary.

One may guess the reason why this project is implemented. Yes, it is because it saves time and effort by saving us the monotony from reading the entire textual data to understand the full passage. Technology has advanced to such an extent that in all fields it has one job and that is to reduce human effort and labour, thus this project was researched on and implemented to save time for the user who may use it to summarize really long case studies into a limited number of pages. This not only allows people to cut down on the reading necessary but also frees up time to read and understand otherwise overlooked written works.

**Chapter 2: Proposed Method with Flow Diagram**

Following methodology is adopted to implement the Text Summarizer :

* First we import the required libraries- nltk, tkinter and pillow.
* Then Cleaning of the Text is done by removing the special characters and white spaces - Converting the text file into individual sentences which will be our document- def clean\_text(file\_name):
* Then find Number of Words in Each Sentence - Number of words in each sentence - def cnt\_in\_sent(sentences):
* Creation of List of Frequencies for each Word in all Documents def freq\_dict(sentences):
* Calculation of TF and IDF Values - The first function calculates the term frequency of words in each document (here, sentences). The second function calculates the inverse document frequency for each word in the sentences- def calc\_TF(text\_data, freq\_list):
* Calculation of TF-IDF Values - Computing the TF-IDF scores for each term- Def sent\_scores(tfidf\_scores, sentences, text\_data);
* Ranking all the Documents - We calculate the score for each sentence based on the TF-IDF value for each word in the sentence
* Generation of the Summary - The threshold is calculated as a linear function of the average of the TF-IDF scores

****

Flow diagram of the entire methodology.

**Chapter 3: Modules Description**

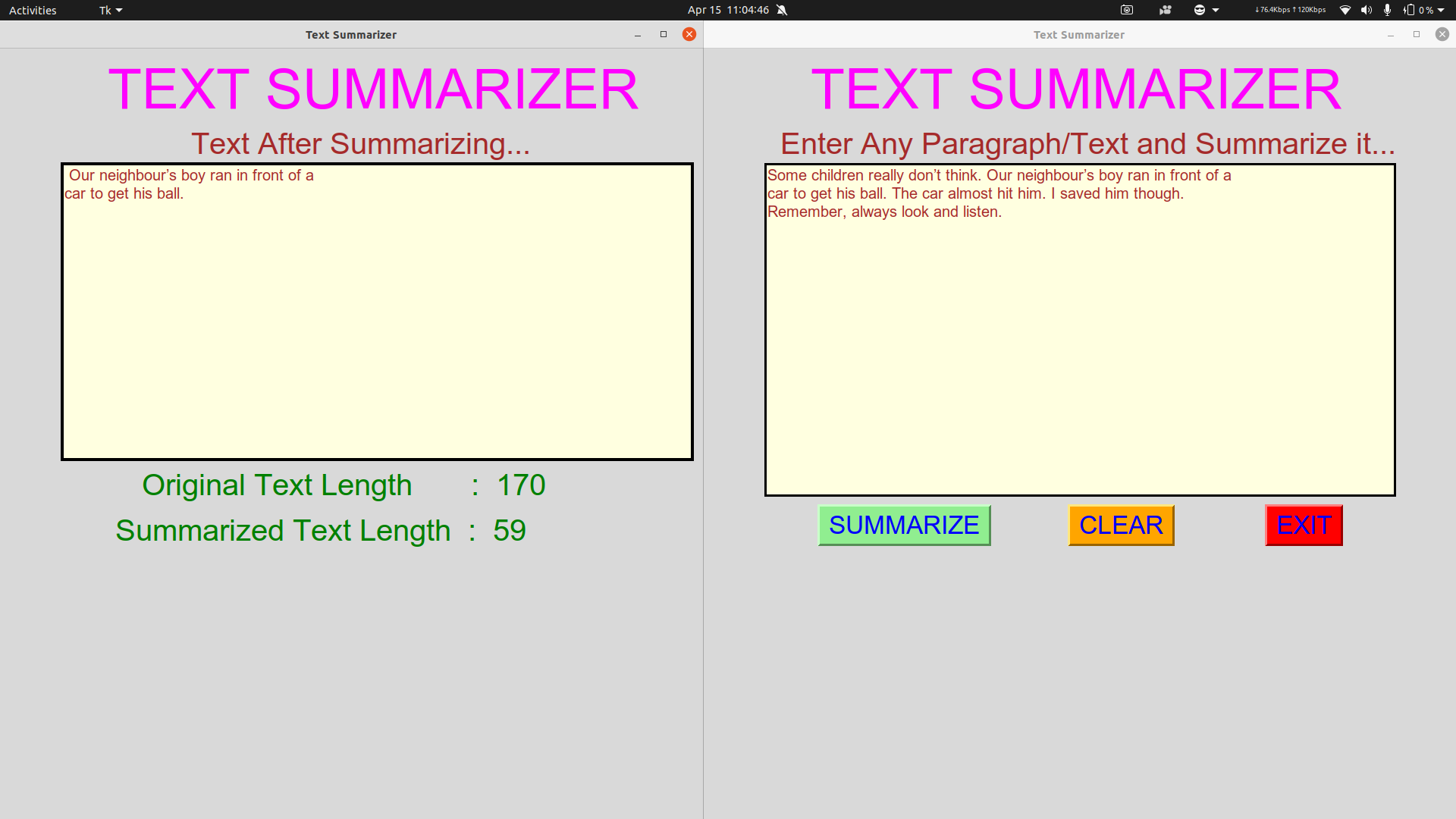
The different modules with their respective contributors are mentioned below.

* Cleaning of the Text - Hari
* Finding Number of Words - Hari
* Creation of List of Frequencies for each Word - Shubhank
* Calculation of TF and IDF Values - Vardhit
* Ranking all the Documents - Vardhit
* Generation of the Summary - Shubhank

**Chapter 5: Implementation requirements**

* python3 needs to be installed in the system
* Jupyter notebook/ vscode needs to be installed in the system
* The required libraries to be installed are tkinter, nltk and pillow
* git also needs to be installed for version control along with a github account.

**Output Screenshot**



**Conclusion**

Text Summarization has been shown to be useful for Natural Language Processing tasks such as Question Answering or Text Classification and other related fields of computer science such as Information Retrieval. And the access time for information searching will be improved. Maximizing your efficiency by minimizing the time you spend reading can have a dramatic impact on productivity. Whether you’re reading textbooks, reports, or academic journals, the power of natural language processing with Python can reduce the time you spend without diluting the quality of information by just providing you a shorter version of the original text. The field of text summarization is experiencing rapid growth, and specialized tools are being developed to tackle more focused summarization tasks. With open-source software and word embedding packages becoming widely available, users are stretching the use case of this technology. It is only a matter of time that such summarizers get integrated so well that they create summaries indistinguishable from those written by humans.

**References**

* <https://www.geeksforgeeks.org/project-idea-text-summarizer/>
* <https://medium.com/spidernitt/how-to-build-a-text-summarizer-from-scratch-1a68e39558c4>

**Appendix A – Source Code:**

**# imported necessary library**

**import tkinter**

**from tkinter import \***

**import tkinter as tk**

**import tkinter.messagebox as mbox**

**from PIL import Image, ImageTk**

**import nltk**

**from nltk.corpus import stopwords**

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

**nltk.download('stopwords')**

**nltk.download('punkt')**

**# Main Window**

**frame = Tk()**

**frame.title('Text Summarizer')**

**frame.geometry('1000x700')**

**# frame.configure(bg = "white")**

**# image on the main window**

**path = "Images/front.jpg"**

**# Creates a Tkinter-compatible photo image, which can be used everywhere Tkinter expects an image object.**

**img1 = ImageTk.PhotoImage(Image.open(path))**

**# The Label widget is a standard Tkinter widget used to display a text or image on the screen.**

**panel = tk.Label(frame, image = img1)**

**panel.place(x = 110, y = 120)**

**# starting label**

**start1 = Label(frame, text='TEXT SUMMARIZER', font=("Arial", 55,"underline"),fg="magenta")**

**start1.place(x=140,y=10)**

**def start\_fun():**

**frame.destroy()**

**# creating an exit button**

**prevB = Button(frame, text='START', command=start\_fun, font=("Arial", 25), bg = "light green", fg = "blue", borderwidth=3, relief="raised")**

**prevB.place(x = 150, y = 600)**

**# defined exit\_win function, to show a exit dialog box when tried to exit**

**def exit\_win():**

**if mbox.askokcancel("Exit", "Do you want to exit?"):**

**frame.destroy()**

**# creating an exit button**

**prevB = Button(frame, text='EXIT', command=exit\_win, font=("Arial", 25), bg = "red", fg = "blue", borderwidth=3, relief="raised")**

**prevB.place(x = 740, y = 600)**

**# this is done to show the exit dialog box when tried to exit from the main window, using the top-roght close button of titlebar**

**frame.protocol("WM\_DELETE\_WINDOW", exit\_win)**

**frame.mainloop()**

**# created main window**

**window = Tk()**

**window.geometry("1000x700")**

**window.title("Text Summarizer")**

**s = ""**

**def summarize\_fun():**

**global s**

**s = ""**

**text = text\_enter.get("1.0", "end-1c")**

**# Tokenizing the text**

**stopWords = set(stopwords.words("english"))**

**words = word\_tokenize(text)**

**# Creating a frequency table to keep the score of each word**

**freqTable = dict()**

**for word in words:**

**word = word.lower()**

**if word in stopWords:**

**continue**

**if word in freqTable:**

**freqTable[word] += 1**

**else:**

**freqTable[word] = 1**

**# Creating a dictionary to keep the score of each sentence**

**sentences = sent\_tokenize(text)**

**sentenceValue = dict()**

**for sentence in sentences:**

**for word, freq in freqTable.items():**

**if word in sentence.lower():**

**if sentence in sentenceValue:**

**sentenceValue[sentence] += freq**

**else:**

**sentenceValue[sentence] = freq**

**sumValues = 0**

**for sentence in sentenceValue:**

**sumValues += sentenceValue[sentence]**

**# Average value of a sentence from the original text**

**average = int(sumValues / len(sentenceValue))**

**# Storing sentences into our summary.**

**summary = ''**

**for sentence in sentences:**

**if (sentence in sentenceValue) and (sentenceValue[sentence] > (1.2 \* average)):**

**summary += " " + sentence**

**# created main window**

**window1 = Tk()**

**window1.geometry("1000x700")**

**window1.title("Text Summarizer")**

**# top label**

**start1 = tk.Label(window1,text="TEXT SUMMARIZER", font=("Arial", 55), fg="magenta") # same way bg**

**start1.place(x=140, y=10)**

**# second label**

**sec1 = tk.Label(window1,text="Text After Summarizing...", font=("Arial", 30), fg="brown") # same way bg**

**sec1.place(x=250, y=100)**

**# created text area**

**text\_enter1 = tk.Text(window1, height=16, width=75, font=("Arial", 15), bg="light yellow", fg="brown", borderwidth=3,relief="solid")**

**text\_enter1.place(x=80, y=150)**

**text\_enter1.delete("1.0", "end")**

**text\_enter1.insert(END, summary)**

**# mbox.showinfo("Success", "Paragraph or Text summarized successfully.\n\nLength of Original text : " + str(len(text)) + "\n\nLength of Summarized text : " + str(len(summary)))**

**# second label**

**sec1 = tk.Label(window1, text="Original Text Length : " + str(len(text)), font=("Arial", 30), fg="green") # same way bg**

**sec1.place(x=185, y=550)**

**# second label**

**sec1 = tk.Label(window1, text="Summarized Text Length : " + str(len(summary)), font=("Arial", 30), fg="green") # same way bg**

**sec1.place(x=150, y=610)**

**# def close\_new():**

**# window1.destroy()**

**#**

**# # created close button**

**# closeb = Button(window1, text="CLOSE", command=close\_new, font=("Arial", 25), bg="orange", fg="blue",borderwidth=3, relief="raised")**

**# closeb.place(x=400, y=600)**

**# top label**

**start1 = tk.Label(text = "TEXT SUMMARIZER", font=("Arial", 55), fg="magenta") # same way bg**

**start1.place(x = 140, y = 10)**

**# second label**

**sec1 = tk.Label(text="Enter Any Paragraph/Text and Summarize it...", font=("Arial", 30), fg="brown") # same way bg**

**sec1.place(x=100, y=100)**

**# created text area**

**text\_enter = tk.Text(window, height=18, width=75, font=("Arial", 15), bg="light yellow", fg="brown", borderwidth=3,relief="solid")**

**text\_enter.place(x=80, y=150)**

**# function for clearing the entry box**

**def clear\_text():**

**text\_enter.delete("1.0", END)**

**# created check button**

**checkb = Button(window, text="SUMMARIZE",command=summarize\_fun,font=("Arial", 25), bg = "light green", fg = "blue", borderwidth=3, relief="raised")**

**checkb.place(x =150 , y =600 )**

**# created clear button**

**clearb = Button(window,text="CLEAR", command=clear\_text, font=("Arial", 25), bg="orange", fg="blue",borderwidth=3, relief="raised")**

**clearb.place(x=480, y=600)**

**# function for exiting**

**def exit\_win():**

**if mbox.askokcancel("Exit", "Do you want to exit?"):**

**window.destroy()**

**# created exit button**

**exitb = Button(window, text="EXIT",command=exit\_win,font=("Arial", 25), bg = "red", fg = "blue", borderwidth=3, relief="raised")**

**exitb.place(x =740 , y =600 )**

**window.protocol("WM\_DELETE\_WINDOW", exit\_win)**

**window.mainloop()**

**Appendix B – GitHub Profile :** <https://github.com/shubhanksargam>

**Link for the Project**:

<https://github.com/shubhanksargam/Text-Summarizer>