## A Project Report

**on**

**NEWS ARTICLE SUMMARIZATION**

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

**BACHELOR OF TECHNOLOGY**

**in**

**COMPUTER SICENCE AND ENGINEERING**

**by**

**16WH1A0595 Ms. P.MYTHREYE**

**16WH1A0594 Ms. N.SRAVANI PRAGNYA**

**17WH5A0506 Ms. R.SAI PRIYANKA**

**under the esteemed guidance of**

**Mr. U.CHANDRASEKHAR**

**Associate Professor**

****

**Department of Computer Science and Engineering**

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**

**May, 2020**

**DECLARATION**

We hereby declare that the work presented in this project entitled **“NEWS ARTICLE SUMMARIZATION**”submitted towards completion of Project Work in IV year of B.Tech., CSE at ‘BVRIT HYDERABAD College of Engineering For Women**’**, Hyderabad is an authentic record of our original work carried out under the guidance of Mr. U.Chandrasekhar, Associate Professor, Department of CSE.

Sign. with date:

**Ms. P.MYTHREYE**

**(16WH1A0595)**

Sign. with date:

**Ms. N.SRAVANI PRAGNYA**

**(16WH1A0594)**

Sign. with date:

**Ms. R.SAI PRIYANKA**

**(17WH5A0506)**

BVRIT HYDERABAD

College of Engineering for Women

(NBA Accredited – EEE, ECE, CSE and IT)

**(Approved by AICTE, New Delhi and Affiliated to JNTUH, Hyderabad)**

**Bachupally, Hyderabad – 500090**

**Department of Computer Science and Engineering**

****

###### Certificate

This is to certify that the Project Work report on **“NEWS ARTICLE SUMMARIZATION”** is a bonafide work carried out by Ms.P.MYTHREYE (16WH1A0595) ; Ms.N.SRAVANI PRAGNYA (16WH1A0594) ; Ms.R.SAI PRYANKA (17WH5A0506) in the partial fulfillment for the award of B.Tech. degree in **Computer Science and Engineering, BVRIT HYDERABAD College of Engineering for Women, Bachupally, Hyderabad**, affiliated to Jawaharlal Nehru Technological University Hyderabad, Hyderabad under my guidance and supervision.

The results embodied in the project work have not been submitted to any other University or Institute for the award of any degree or diploma.

**Head of the Department Guide**

**Dr. Ch.Srinivasulu Mr.U.Chandrasekhar**

**Professor and HoD, Associate Professor**

**Department of CSE**

**External Examiner**

**Acknowledgements**

We would like to express our sincere thanks to **Dr. K V N Sunitha, Principal**, **BVRIT HYDERABAD College of Engineering for Women**, for providing the working facilities in the college.

Our sincere thanks and gratitude to our **Dr. Ch.Srinivasulu, Professor**, Department of CSE, **BVRIT HYDERABAD College of Engineering for Women** for all the timely support and valuable suggestions during the period of our project.

We are extremely thankful and indebted to our internal guide, **Mr. U.Chandrasekhar, Associate Professor**, Department of CSE**, BVRIT HYDERABAD College of Engineering for Women** for his constant guidance, encouragement and moral support throughout the project.

Finally, we would also like to thank our Project Coordinator, all the faculty and staff of **CSE** Department who helped us directly or indirectly, parents and friends for their cooperation in completing the project work.

**Ms. P.MYTHREYE**

**(16WH1A0595)**

**Ms. N.SRAVANI PRAGNYA**

**(16WH1A0594)**

**Ms.R.SAI PRIYANKA**

**(17WHA0506)**

**Contents**

|  |  |  |
| --- | --- | --- |
| **S.No.** | **Topic** | **Page No.** |
|  | Abstract | i |
|  | List of Figures | ii |
| 1. | Introduction | 9 |
|  | 1.1 Objectives | 9 |
|  | 1.2 Methodology | 9 |
|  | 1.2.1 proposed system | 9 |
|  | 1.2.2 Organization of Project  1.2.3 Dataset  1.2.4 Goodness measure | 9  10  10 |
|  |  |  |
| 2. | Theoretical Analysis of the proposed project | 12 |
|  | 2.1 Requirements Gathering | 12 |
|  | 2.1.1 Software Requirements | 12 |
|  | 2.1.2 Hardware Requirements | 12 |
|  | 2.1.3 Technologies Description | 12 |
| 3. | Design | 18 |
|  | 3.1 Introduction | 18 |
|  | 3.2 Architecture Diagram | 19 |
|  | 3.3 UML Diagrams | 21 |
|  | 3.3.1 Use Case Diagram | 21 |
|  | 3.3.2 Sequence Diagram | 22 |
|  | 3.3.3 Activity Diagram | 23 |
|  |
| 4. | Implementation | 25 |
|  | 4.1 Coding | 25 |
|  | 4.2 Testing | 37 |
|  | 4.2.1 Testing Strategies | 37 |
|  | 4.3 Test Cases | 39 |
|  | 4.4 Input Screenshots  4.5 Output Screenshots  4.6 Result | 40  41  43 |
| 5.  6. | Conclusion and Future Scope  References | 45  46 |
|  |  |  |

**Abstract**

With the availability of World Wide Web in every corner of the world these days, the amount of information on the internet is growing at an exponential rate. However, given the hectic schedule of people and the immense amount of information available, there is increase in need for information abstraction or summarization. Text summarization presents the user a shorter version of text with only vital information and thus helps him to understand the text in shorter amount of time.The goal of news article summarization is to condense the news article into a shorter version and preserve important contents. The main advantage of summarization lies in the fact that it reduces user's time in searching the important details in the articles. When humans summarize an article, they first read and understand the article or document and then capture the important points. They then use these important points to generate their own sentences to communicate the gist of the article. Even though the quality of summary generated might be excellent, manual summarization is a time consuming process. Hence, the need for automatic summarizers is quite apparent. The most important task in extractive text summarization is choosing the important sentences that would appear in the summary. Identifying such sentences is a truly challenging task. Currently, automatic text summarization has applications in several areas such as news articles, emails, research papers and online search engines to receive summary of results found.

**List of Figures**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Figure Description** | **Page.No.** |
| 1 | Architecture Diagram | 19 |
| 2 | UseCase Diagram | 21 |
| 3 | Sequence Diagram | 22 |
| 4 | Activity Diagram | 23 |
| 5 | TestCases | 39 |
| 6 | Uploading url of news website | 40 |
| 7 | Uploading news article | 40 |
| 8 | Output summary | 40 |
| 9 | Output summary using NLTK | 41 |
| 10 | Output summary using SUMY LEXRANK | 42 |
| 11 | Output summary using SpaCy | 42 |

**1 INTRODUCTION**

* 1. **Objective**

With the availability of World Wide Web in every corner of the world these days, the amount of information on the internet is growing at an exponential rate. However, given the hectic schedule of people and the immense amount of information available, there is increase in need for information abstraction or summarization. Text summarization presents the user a shorter version of text with only vital information and thus helps him to understand the text in shorter amount of time.The goal of news article summarization is to condense the news article into a shorter version and preserve important contents. The main advantage of summarization lies in the fact that it reduces user's time in searching the important details in the articles. When humans summarize an article, they first read and understand the article or document and then capture the important points. They then use these important points to generate their own sentences to communicate the gist of the article. Even though the quality of summary generated might be excellent, manual summarization is a time consuming process. Hence, the need for automatic summarizers is quite apparent. The most important task in extractive text summarization is choosing the important sentences that would appear in the summary. Identifying such sentences is a truly challenging task. Currently, automatic text summarization has applications in several areas such as news articles, emails, research papers and online search engines to receive summary of results found.

**1.2 Methodology**

**1.2.1 Proposed System :-**

**News Article Summarization** Involves Extractive summarization using NLTK that extracts keywords or key sentences from the original article without changing the sentences. Then, these extracted sentences can be used to form a summary of an article.

**1.2.2 Organisation of Project :-**

In the extractive summarization, the summarizer takes input as text/url and pre-processing is done in-order to remove the stop words . Then tokenization is done to find the terms of the input text.

**Step 1:** Weights are assigned to the tokens. The term weight is calculated as follows:

Wt = frequency of a token

**Step 2:** Now maximum weight of the token is considered after finding maximum weight. The term weighted frequency of an article is calculated as follows:

Wtf = frequency of token/maximum frequency of token

**Step 3:** In this step, the frequencies are connecting in place of corresponding words in sentence and sum of it is found. The ranks are found based on the weighted frequency. The sentences are sorted based on their Weighted frequency ranks like highest rank to lowest. The sentences are arranged in descending order.

Step 4: Finally, summarizer will extract sentences whose rank is highest form the input article and the those sentences form the summary.

**1.2.3 Dataset**

The dataset used is a BBC news dataset which consists of news articles with their summaries. This dataset consists of 5 categories of news articles. Categories are as follows:

|  |
| --- |
| **CATEGORY** |
| Tech |
| Politics |
| Sport |
| Entertainment |
| Business |

**1.2.4 Goodness Measure**

**ROUGE SCORE**

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. It is essentially a set of metrics for evaluating automatic summarization of texts as well as machine translations.

It works by comparing an **automatically produced summary** or **translation** against a set of **reference summaries** (typically human-produced). Let’s say that we have the following system and reference summaries:

**System Summary (what the machine produced):**

the cat was found under the bed

**Reference Summary (gold standard — usually by humans):**

the cat was under the bed

Simply put, recall (in the context of ROUGE) refers to how much of the **reference summary** the **system summary** is recovering or capturing. If we are just considering the individual words, it can be computed as:

Number of overlapping words/Total number of words in reference summary

In this example, the recall would thus be:

Recall = 6/6 = 1.0

This means that all the words in the **reference summary** have been captured by the **system summary**

**2.THEORETICAL ANALYSIS OF THE PROPOSED PROJECT**

* 1. **Requirements Gathering**
     1. **Software Requirements:**
* Programming Languages : Python 3.6
* Graphical User Interface : HTML, Javascript
* Tool : Anaconda Prompt
* Framework : Flask Framework
  + 1. **Hardware Requirements:**
* Operating System : Windows 10
* Processor : (Minimum) Any Intel or AMDx86-64
* RAM : 4GB
* Hard Disk : 100GB
  + 1. **Technologies Description:**
* **Python :**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

* Python is Interpreted − Python is processed at runtime by the interpreter.
* You do not need to compile your program before executing it. This is similar to PERL and PHP.
* Python is Interactive − you can actually sit at a Python prompt and interact

with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

**Advantages of Python:**

1. **Extensive Libraries**

Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don‟t have write the complete code for that manually.

1. **Extensible**

As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

1. **Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.

1. **Improved Productivity**

The language‟s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. You can write less and get more things done.

1. **IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

1. **Simple and Easy**

When working with Java, you may have to create a class to print „Hello World‟. But in Python, just a print statement will do. It is also quite easy to learn, understand,and code. This is why when people pick up Python; they have a hard time adjusting to other more verbose languages like Java.

1. **Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

1. **Object-Oriented**

This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.

* **HTML**:

HTML stands for Hypertext Markup Language, and it is the most widely used

language to write Web Pages.

* Hypertext refers to the way in which Web pages (HTML documents) are

linked together. Thus, the link available on a webpage is called Hypertext.

* As its name suggests, HTML is a Markup Language which means you use

HTML to simply “mark-up” a text document with tags that tell a Web browser how to structure it to display.

Originally, HTML was developed with the intent of defining the structure of

documents like headings, paragraphs, lists, and so forth to facilitate the sharing of

scientific information between researchers.

Now, HTML is being widely used to format web pages with the help of different tags available in HTML language**.**

* **Flask Framework :**

**Flask** is a micro [web framework](https://en.wikipedia.org/wiki/Web_framework) written in [Python](https://en.wikipedia.org/wiki/Python_(programming_language)). It is classified as a [microframework](https://en.wikipedia.org/wiki/Microframework) because it does not require particular tools or libraries.It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object-relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools. Extensions are updated far more frequently than the core Flask program.

Applications that use the Flask framework includes [Pinterest](https://en.wikipedia.org/wiki/Pinterest) and [LinkedIn](https://en.wikipedia.org/wiki/LinkedIn).

Features :

* Development server and debugger
* Integrated support for unit testing
* RESTful request dispatching
* Support for secure cookies (client side sessions)
* **Web Scraping:**

**Web scraping**, **web harvesting**, or **web data extraction** is [data scraping](https://en.wikipedia.org/wiki/Data_scraping) used for [extracting data](https://en.wikipedia.org/wiki/Data_extraction) from [websites](https://en.wikipedia.org/wiki/Website). Web scraping software may access the World Wide Web directly using the [Hypertext Transfer Protocol](https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol), or through a web browser.

Web scraping a web page involves fetching it and extracting from it. Fetching is the downloading of a page (which a browser does when you view the page). Therefore, web crawling is a main component of web scraping, to fetch pages for later processing. Once fetched, then extraction can take place. The content of a page may be [parsed](https://en.wikipedia.org/wiki/Parsing), searched, reformatted, its data copied into a spreadsheet, and so on. Web scrapers typically take something out of a page, to make use of it for another purpose somewhere else. An example would be to find and copy names and phone numbers, or companies and their URLs, to a list (contact scraping).

Web scraping is used for [contact scraping](https://en.wikipedia.org/wiki/Contact_scraping), and as a component of applications used for [web indexing](https://en.wikipedia.org/wiki/Web_indexing), [web mining](https://en.wikipedia.org/wiki/Web_mining) and [data mining](https://en.wikipedia.org/wiki/Data_mining), online price change monitoring and [price comparison](https://en.wikipedia.org/wiki/Comparison_shopping_website), product review scraping (to watch the competition), gathering real estate listings, weather data monitoring, [website change detection](https://en.wikipedia.org/wiki/Change_detection_and_notification), research, tracking online presence and reputation, [web mashup](https://en.wikipedia.org/wiki/Web_mashup) and, [web data integration](https://en.wikipedia.org/wiki/Web_data_integration).

* **NLTK:**

The **Natural Language Toolkit**, or more commonly **NLTK**, is a suite of [libraries](https://en.wikipedia.org/wiki/Library_(computer_science)) and programs for symbolic and statistical [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing) (NLP) for English written in the [Python programming language](https://en.wikipedia.org/wiki/Python_(programming_language)). NLTK includes graphical demonstrations and sample data.

NLTK is intended to support research and teaching in [NLP](https://en.wikipedia.org/wiki/Natural_Language_Processing) or closely related areas, including empirical [linguistics](https://en.wikipedia.org/wiki/Linguistics), [cognitive science](https://en.wikipedia.org/wiki/Cognitive_science), [artificial intelligence](https://en.wikipedia.org/wiki/Artificial_intelligence), [information retrieval](https://en.wikipedia.org/wiki/Information_retrieval), and [machine learning](https://en.wikipedia.org/wiki/Machine_learning). NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems.

NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

* **Spacy:**

**spaCy** is an [open-source](https://en.wikipedia.org/wiki/Open-source_software) software library for advanced [natural language processing](https://en.wikipedia.org/wiki/Natural_language_processing), written in the programming languages [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) and [Cython](https://en.wikipedia.org/wiki/Cython). Unlike [NLTK](https://en.wikipedia.org/wiki/Natural_Language_Toolkit), which is widely used for teaching and research, spaCy focuses on providing software for production usage. As of version 1.0, spaCy also supports [deep learning](https://en.wikipedia.org/wiki/Deep_learning) workflows that allow connecting statistical models trained by popular [machine learning](https://en.wikipedia.org/wiki/Machine_learning) libraries like [TensorFlow](https://en.wikipedia.org/wiki/TensorFlow), [Keras](https://en.wikipedia.org/wiki/Keras), [Scikit-learn](https://en.wikipedia.org/wiki/Scikit-learn) or [PyTorch](https://en.wikipedia.org/wiki/PyTorch).

spaCy's machine learning library, Thinc, is also available as a separate [open-source](https://en.wikipedia.org/wiki/Open-source_software) [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) library.[[10]](https://en.wikipedia.org/wiki/SpaCy#cite_note-10) It features [convolutional neural network](https://en.wikipedia.org/wiki/Convolutional_neural_network) models for [part-of-speech tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging), dependency parsing and [named entity recognition](https://en.wikipedia.org/wiki/Named-entity_recognition), as well as API improvements around training and updating models, and constructing custom processing pipelines.

Features :

* Non-destructive [tokenization](https://en.wikipedia.org/wiki/Tokenization_(lexical_analysis))
* [Named entity recognition](https://en.wikipedia.org/wiki/Named-entity_recognition)
* "Alpha tokenization" support for over 50 languages
* [Statistical models](https://en.wikipedia.org/wiki/Statistical_model) for 11 languages[[12]](https://en.wikipedia.org/wiki/SpaCy#cite_note-12)
* Pre-trained [word vectors](https://en.wikipedia.org/wiki/Word_embedding)
* [Part-of-speech tagging](https://en.wikipedia.org/wiki/Part-of-speech_tagging)
* Labelled [dependency](https://en.wikipedia.org/wiki/Dependency_grammar) parsing
* Syntax-driven [sentence segmentation](https://en.wikipedia.org/wiki/Sentence_boundary_disambiguation)
* **JavaScript :**

JavaScript is a dynamic computer programming language. It is lightweight and most commonly used as a part of web pages, whose implementations allow client-side script to interact with the user and make dynamic pages. It is an interpreted

programming language with object-oriented capabilities.

JavaScript was first known as LiveScript, but Netscape changed its name to

JavaScript, possibly because of the excitement being generated by Java. JavaScript made its first appearance in Netscape 2.0 in 1995 with the name LiveScript. The general-purpose core of the language has been embedded in Netscape, Internet Explorer, and other web browsers.

The ECMA-262 Specification defined a standard version of the core JavaScript

language.

* JavaScript is a lightweight, interpreted programming language.
* Designed for creating network-centric applications.
* Complementary to and integrated with Java.
* Complementary to and integrated with HTML.
* Open and cross-platform**.**
* **Beautiful Soap :**

**Beautiful Soup** is a [Python](https://en.wikipedia.org/wiki/Python_(programming_language)) package for parsing [HTML](https://en.wikipedia.org/wiki/HTML) and [XML](https://en.wikipedia.org/wiki/XML) documents (including having malformed markup, i.e. non-closed tags, so named after [tag soup](https://en.wikipedia.org/wiki/Tag_soup)). It creates a parse tree for parsed pages that can be used to extract data from HTML, which is useful for [web scraping](https://en.wikipedia.org/wiki/Web_scraping).

**3 DESIGN**

**3.1 Introduction**

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software. The importance can be stated with a single word “Quality”. Design is the place where quality is fostered in software development. Design provides us with representations of software that can assess for quality. Design is the only way that we can accurately translate a customer’s view into a finished software product or system. Software design serves as a foundation for all the software engineering steps that follow. Without a strong design we risk building an unstable system – one that will be difficult to test, one whose quality cannot be assessed until the last stage. During design, progressive refinement of data structure, program structure, and procedural details are developed reviewed and documented. System design can be viewed from either technical or project management perspective. From the technical point of view, design is comprised of four activities – architectural design, data structure design, interface design and procedural design.

88.

**3.2 Architecture**

Web applications are by nature distributed applications, meaning that they are programs that run on more than one computer and communicate through network or server. Specifically, web applications are accessed with a web browser and are popular because of the ease of using the browser as a user client. For the enterprise, software on potentially thousands of client computers is a key reason for their popularity. Web applications are used for web mail, online retail sales, discussion boards, weblogs, online banking, and more. One web application can be accessed and used by millions of people.

Like desktop applications, web applications are made up of many parts and often contain mini programs and some of which have user interfaces. In addition, web applications frequently require an additional markup or scripting language, such as HTML, CSS, or JavaScript programming language. Also, many applications use only the Python programming language, which is ideal because of its versatility.

News Article

Natural Language Processing

Calculating word frequency

Calculating sentence score based on word frequency

Choosing the top ‘n’ sentences

Output summary

**Fig 1:Architecture Diagram**

**3.3 UML Diagrams**

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

**3.3.1 Use Case Diagram**

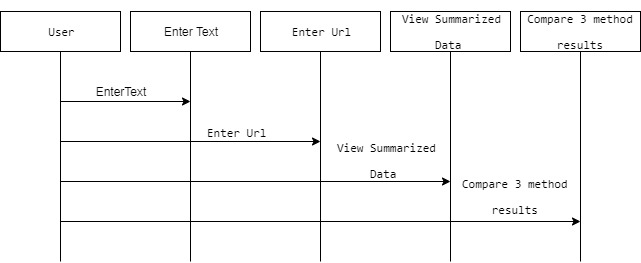
A use case diagram at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.

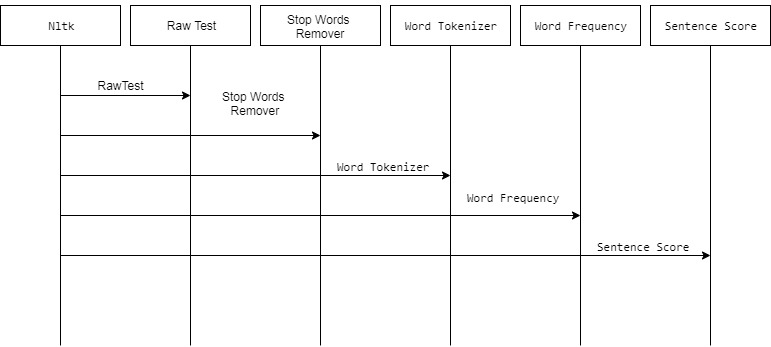
user

**Fig 2:Use Case Diagram**

**3.3.2 Sequence Diagram**

A sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.





**Fig 3:Sequence Diagram**

**3.3.3 Activity Diagram**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

Sentence Score

Word Frequency

Tokenization

Text Preprocessing

Input text or URL

Output Summary

**Fig 4:Activity Diagram**

**4.IMPLEMENTATION**

**4.1 Coding**

**Sample Code**

**App.py**

from \_\_future\_\_ import unicode\_literals

from flask import Flask,render\_template,url\_for,request

from spacy\_summarization import text\_summarizer

from gensim.summarization import summarize

from nltk\_summarization import nltk\_summarizer

import time

import spacy

nlp = spacy.load('en\_core\_web\_sm')

app = Flask(\_\_name\_\_)

# Web Scraping Pkg

from bs4 import BeautifulSoup

# from urllib.request import urlopen

from urllib.request import urlopen

# Sumy Pkg

from sumy.parsers.plaintext import PlaintextParser

from sumy.nlp.tokenizers import Tokenizer

from sumy.summarizers.lex\_rank import LexRankSummarizer

# Sumy

def sumy\_summary(docx):

parser = PlaintextParser.from\_string(docx,Tokenizer("english"))

lex\_summarizer = LexRankSummarizer()

summary = lex\_summarizer(parser.document,3)

summary\_list = [str(sentence) for sentence in summary]

result = ' '.join(summary\_list)

return result

# Reading Time

def readingTime(mytext):

total\_words = len([ token.text for token in nlp(mytext)])

estimatedTime = total\_words/200.0

return estimatedTime

# Fetch Text From Url

def get\_text(url):

page = urlopen(url)

soup = BeautifulSoup(page)

fetched\_text = ' '.join(map(lambda p:p.text,soup.find\_all('p')))

return fetched\_text

@app.route('/')

def index():

return render\_template('index.html')

@app.route('/analyze',methods=['GET','POST'])

def analyze():

start = time.time()

if request.method == 'POST':

rawtext = request.form['rawtext']

final\_reading\_time = readingTime(rawtext)

final\_summary = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary)

end = time.time()

final\_time = end-start

return render\_template('index.html',ctext=rawtext,final\_summary=final\_summary,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time)

@app.route('/analyze\_url',methods=['GET','POST'])

def analyze\_url():

start = time.time()

if request.method == 'POST':

raw\_url = request.form['raw\_url']

rawtext = get\_text(raw\_url)

final\_reading\_time = readingTime(rawtext)

final\_summary = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary)

end = time.time()

final\_time = end-start

return render\_template('index.html',ctext=rawtext,final\_summary=final\_summary,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time)

@app.route('/compare\_summary')

def compare\_summary():

return render\_template('compare\_summary.html')

@app.route('/comparer',methods=['GET','POST'])

def comparer():

start = time.time()

if request.method == 'POST':

rawtext = request.form['rawtext']

final\_reading\_time = readingTime(rawtext)

final\_summary\_spacy = text\_summarizer(rawtext)

summary\_reading\_time = readingTime(final\_summary\_spacy)

# Gensim Summarizer

final\_summary\_gensim = summarize(rawtext)

summary\_reading\_time\_gensim = readingTime(final\_summary\_gensim)

# NLTK

final\_summary\_nltk = nltk\_summarizer(rawtext)

summary\_reading\_time\_nltk = readingTime(final\_summary\_nltk)

# Sumy

final\_summary\_sumy = sumy\_summary(rawtext)

summary\_reading\_time\_sumy = readingTime(final\_summary\_sumy)

end = time.time()

final\_time = end-start

return render\_template('compare\_summary.html',ctext=rawtext,final\_summary\_spacy=final\_summary\_spacy,final\_summary\_gensim=final\_summary\_gensim,final\_summary\_nltk=final\_summary\_nltk,final\_time=final\_time,final\_reading\_time=final\_reading\_time,summary\_reading\_time=summary\_reading\_time,summary\_reading\_time\_gensim=summary\_reading\_time\_gensim,final\_summary\_sumy=final\_summary\_sumy,summary\_reading\_time\_sumy=summary\_reading\_time\_sumy,summary\_reading\_time\_nltk=summary\_reading\_time\_nltk)

@app.route('/about')

def about():

return render\_template('index.html')

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

**nltk\_summarization.py**

import nltk

from nltk.corpus import stopwords

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

import heapq

def nltk\_summarizer(raw\_text):

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

word\_frequencies = {}

for word in nltk.word\_tokenize(raw\_text):

if word not in stopWords:

if word not in word\_frequencies.keys():

word\_frequencies[word] = 1

else:

word\_frequencies[word] += 1

maximum\_frequncy = max(word\_frequencies.values())

for word in word\_frequencies.keys():

word\_frequencies[word] = (word\_frequencies[word]/maximum\_frequncy)

sentence\_list = nltk.sent\_tokenize(raw\_text)

sentence\_scores = {}

for sent in sentence\_list:

for word in nltk.word\_tokenize(sent.lower()):

if word in word\_frequencies.keys():

if len(sent.split(' ')) < 30:

if sent not in sentence\_scores.keys():

sentence\_scores[sent] = word\_frequencies[word]

else:

sentence\_scores[sent] += word\_frequencies[word]

summary\_sentences = nlargest(7, sentence\_scores, key=sentence\_scores.get)

summary = ' '.join(summary\_sentences)

return summary

**spacy\_summarizer.py**

# NLP Pkgs

import spacy

nlp = spacy.load('en')

# Pkgs for Normalizing Text

from spacy.lang.en.stop\_words import STOP\_WORDS

from string import punctuation

# Import Heapq for Finding the Top N Sentences

from heapq import nlargest

def text\_summarizer(raw\_docx):

raw\_text = raw\_docx

docx = nlp(raw\_text)

stopwords = list(STOP\_WORDS)

# Build Word Frequency # word.text is tokenization in spacy

word\_frequencies = {}

for word in docx:

if word.text not in stopwords:

if word.text not in word\_frequencies.keys():

word\_frequencies[word.text] = 1

else:

word\_frequencies[word.text] += 1

maximum\_frequncy = max(word\_frequencies.values())

for word in word\_frequencies.keys():

word\_frequencies[word] = (word\_frequencies[word]/maximum\_frequncy)

# Sentence Tokens

sentence\_list = [ sentence for sentence in docx.sents ]

# Sentence Scores

sentence\_scores = {}

for sent in sentence\_list:

for word in sent:

if word.text.lower() in word\_frequencies.keys():

if len(sent.text.split(' ')) < 30:

if sent not in sentence\_scores.keys():

sentence\_scores[sent] = word\_frequencies[word.text.lower()]

else:

sentence\_scores[sent] += word\_frequencies[word.text.lower()]

summarized\_sentences = nlargest(7, sentence\_scores, key=sentence\_scores.get)

final\_sentences = [ w.text for w in summarized\_sentences ]

summary = ' '.join(final\_sentences)

print("Original Document\n")

print(raw\_docx)

print("Total Length:",len(raw\_docx))

print('\n\nSummarized Document\n')

print(summary)

print("Total Length:",len(summary))

**index.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1"/>

<title>Summarizer</title>

<!-- CSS -->

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link href="static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>

<link href="static/css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous">

</head>

</head>

<body><div class="navbar-fixed">

<nav class="grey darken-4" role="navigation">

<div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Summarizer</a>

<ul class="right hide-on-med-and-down">

<li><a href="{{url\_for('index')}}">Home</a></li>

<li><a href="{{url\_for('compare\_summary')}}">Compare Summarizers</a></li>

</ul>

<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>

</div>

</nav>

</div>

<ul id="nav-mobile" class="sidenav">

<li><a href="{{url\_for('index')}}">Home</a></li>

<li><a href="{{url\_for('compare\_summary')}}">Compare</a></li>

</ul>

<!--Slider -->

<!-- Start of Main Section -->

<div class="container">

<div class="section">

<!-- Icon Section -->

<div class="row">

<div class="input-field col s12 m10">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>

<form method="POST" action="/analyze">

<textarea name="rawtext" cols="3" rows="5" class="form-control" required="true" placeholder="Enter Text Here"></textarea>

<br/>

<button class="btn btn-small waves-effect waves-light light-blue lighten-1" type="reset">Clear</button>

<button class="btn btn-small waves-effect waves-light purple lighten-1" type="submit">Summarize</button>

</form>

</div>

</div>

</div>

<div class="row">

<div class="input-field col s12 m10">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>

<form method="POST" action="/analyze\_url">

<input type="text" name="raw\_url" placeholder="Enter URL Here" required="true">

<button class="btn btn-small waves-effect waves-light light-blue lighten-1" type="reset">Clear</button>

<button class="btn btn-small waves-effect waves-light purple lighten-1" type="submit">Summarize</button>

</form>

</div>

</div>

</div>

</div>

</div>

<!-- End -->

<!-- Result Display-->

<section class="section section-solutions-about grey darken-2">

<div class="container white-text">

<!-- Icon Section -->

<div class="row">

<div class="col s12 m6">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">group</i></h2>

<h5 class="center">Your Text</h5>

<p>Reading Time: <span style="color:#0091EA;">{{ final\_reading\_time }} min </span></p>

<p class="light">{{ctext}}</p>

<div class="alert alert-info" role="alert"><p>Time Elapsed: <span style="color:#0091EA;">{{ final\_time }} mins </span></p><br/>

</div>

</div>

</div>

<div class="col s12 m6 grey darken-3">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">chrome\_reader\_mode</i></h2>

<h5 class="center">Summarized</h5>

<div class="" role="alert">

<p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time }} min </span></p>

</div>

<p class="light">{{ final\_summary }}</p>

</div>

</div>

</div>

</div>

</section>

<!-- Scripts-->

<script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>

<script src="static/js/materialize.js"></script>

<script src="static/js/init.js"></script>

<!-- Jesse JCharis -->

</body>

</html>

**compare\_summary.html**

<!DOCTYPE html>

<html lang="en">

<head>

<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

<meta name="viewport" content="width=device-width, initial-scale=1"/>

<title>Summarizer</title>

<!-- CSS -->

<link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

<link href="static/css/materialize.css" type="text/css" rel="stylesheet" media="screen,projection"/>

<link href="static/css/style.css" type="text/css" rel="stylesheet" media="screen,projection"/>

<link rel="stylesheet" href="https://use.fontawesome.com/releases/v5.5.0/css/all.css" integrity="sha384-B4dIYHKNBt8Bc12p+WXckhzcICo0wtJAoU8YZTY5qE0Id1GSseTk6S+L3BlXeVIU" crossorigin="anonymous">

</head>

</head>

<body><div class="navbar-fixed">

<nav class="grey darken-4" role="navigation">

<div class="nav-wrapper container">

<a id="logo-container" href="#" class="brand-logo">Summarizer</a>

<ul class="right hide-on-med-and-down">

<li><a href="{{url\_for('index')}}">Home</a></li>

<li><a href="{{url\_for('compare\_summary')}}">Compare Summarizers</a></li>

</ul>

<a href="#" data-target="nav-mobile" class="sidenav-trigger"><i class="material-icons">menu</i></a>

</div>

</nav>

</div>

<ul id="nav-mobile" class="sidenav">

<li><a href="{{url\_for('index')}}">Home</a></li>

<li><a href="{{url\_for('compare\_summary')}}">Compare</a></li>

</ul>

<!--Slider -->

<!-- Start -->

<div class="container">

<div class="section">

<!-- Icon Section -->

<div class="row">

<div class="input-field col s12 m10">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">compare</i></h2>

<form method="POST" action="/comparer">

<textarea name="rawtext" cols="3" rows="5" class="form-control" required="true" placeholder="Enter Text Here"></textarea>

<br/>

<button class="btn btn-small waves-effect waves-light light-blue lighten-1" type="reset">Clear</button>

<button class="btn btn-small waves-effect waves-light purple lighten-1" type="submit">Summarize</button>

</form>

</div>

</div>

</div>

</div>

</div>

<!-- End -->

<!-- Results -->

<section class="section section-solutions-about grey lighten-3 ">

<div class="container">

<div class="row">

<div class="col s12 m8">

<div class="icon-block">

<h2 class="center brown-text"><i class="material-icons">compare</i></h2>

<h5 class="center">Your Text</h5>

<p class="light">{{ctext}}</p>

<div class="alert alert-info" role="alert"><p>Time Elapsed: <span style="color:#0091EA;">{{ final\_time }} </span></p><br/>

<p>Reading Time: <span style="color:#0091EA;">{{ final\_reading\_time }} </span></p>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- SECTION: RESULTS FOR DIFFERENT SUMMARIZERS -->

<section class="section section-solutions-about grey lighten-3 ">

<div class="container">

<div class="row">

<div class="col s12 m12 offset-m1">

<div class="row">

<div class="col s12">

<ul class="tabs">

<li class="tab col s3">

<a href="#tab1" class="blue-text">SpaCy Summarizer</a>

</li>

<li class="tab col s3">

<a href="#tab2" class="blue-text">Gensim Summarizer</a>

</li>

<li class="tab col s3">

<a href="#tab3" class="blue-text">NLTK</a>

</li>

<li class="tab col s3">

<a href="#tab4" class="blue-text">Sumy LexRank</a>

</li>

</ul>

</div>

<div id="tab1" class="col s12">

<h5>SpaCy</h5>

<div class="" ><p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time }} min </span></p></div>

<div class="grey lighten-1"><p style="padding:5px;font-size:14px;">{{ final\_summary\_spacy }}</p></div>

</div>

<div id="tab2" class="col s12">

<h5>Gensim Summarizer</h5>

<div class=""><p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time\_gensim }} min </span></p>

</div> <div class="grey lighten-2"><p style="padding:5px;font-size:14px;">{{ final\_summary\_gensim }}</p></div>

</div>

<div id="tab3" class="col s12">

<h5>NLTK</h5>

<div class=""><p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time\_nltk }} min </span></p>

</div>

<div class="grey lighten-1"><p style="padding:5px;font-size:14px;">{{ final\_summary\_nltk }}</p></div>

</div>

<div id="tab4" class="col s12">

<h5>Sumy Summarizer</h5>

<div class=""><p>Reading Time: <span style="color:#0091EA;">{{ summary\_reading\_time\_sumy}} min </span></p>

</div> <div class="grey lighten-2"><p style="padding:5px;font-size:14px;">{{ final\_summary\_sumy }}</p></div>

</div>

</div>

</div>

</div>

</div>

</section>

<!-- Scripts-->

<script src="https://code.jquery.com/jquery-2.1.1.min.js"></script>

<script src="static/js/materialize.js"></script>

<script src="static/js/init.js"></script>

<!-- Jesse JCharis -->

</body>

</html>

**RougeScore.py**

sys\_summary = set(open("sys\_summary.txt").read().split())

ref\_summary = set(open("ref\_summary.txt").read().split())

l1 = len(sys\_summary)

l2 =len(ref\_summary)

overlapping\_words = sys\_summary.intersection(ref\_summary)

print(len(overlapping\_words)/l2)

**4.2 Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. The following is the description of the testing strategies, which were carried out during the testing period.

**4.2.1 Testing Strategies**

**SYSTEM TESTING**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to user the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**MODULE TESTING**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**INTEGRATION TESTING**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system

## ACCEPTANCE TESTING

## When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

**4.3 Test Cases**

Integration and regression testing strategies are used in this application for testing.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| TestCase ID | Test Scenario | Expected Results | Actual Result | Pass/Fail |
| **1** | Check whether application is working after copying url | Home page should be opened | As Expected | Pass |
| **2** | Check whether summarize option is working | Should get summary of uploaded text/url | As Expected | Pass |
| **3** | Check whether clear option is working | Should clear the uploaded text/url | As Expected | Pass |
| **4** | Check whether compare summarizers option is working | It should open compare summarizers page | As Expected | Pass |
| **5** | Check whether spacy summarizer option is working | Should get summary of uploaded article using spacy technique | As Expected | Pass |
| **6** | Check whether nltk summarizer option is working | Should get summary of uploaded article using nltk technique | As Expected | Pass |
| **7** | Check whether genism summarizer option is working | Should get summary of uploaded article using genism technique | As Expected | Pass |
| **8** | Check whether sumy lexrank summarizer option is working | Should get summary of uploaded article using sumy lexrank technique | As Expected | Pass |

**Fig 5:Test Cases**

## 4.4 Input Screenshots

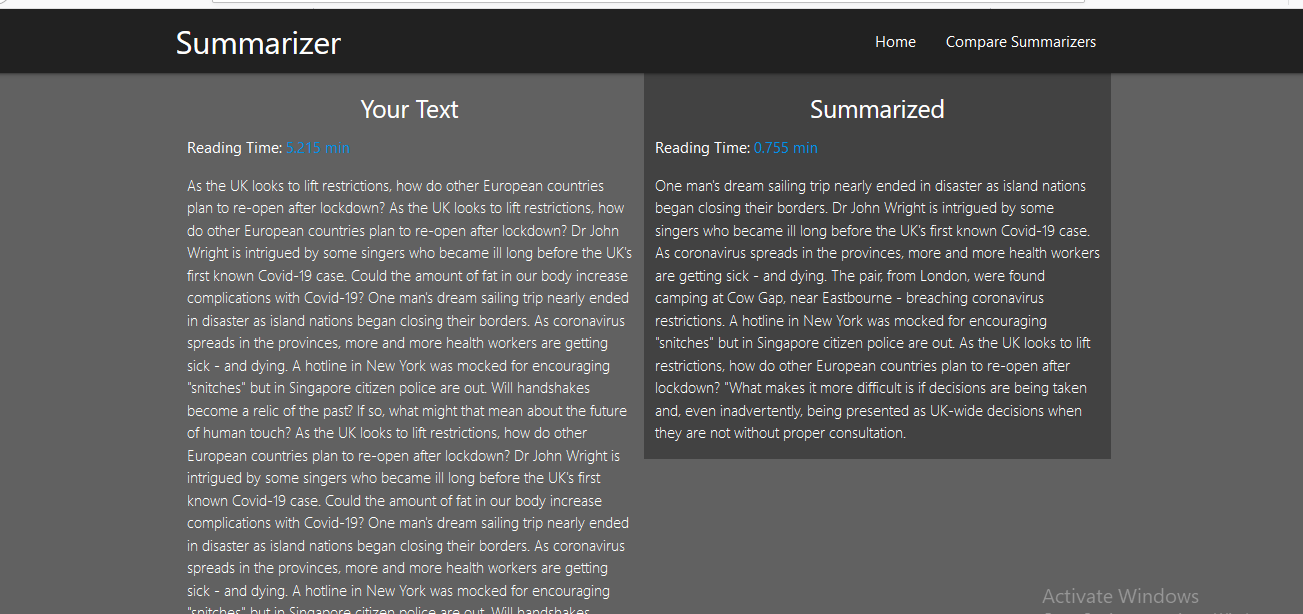
## Screenshot (47).png

**Fig 6:Uploading URL of news website**

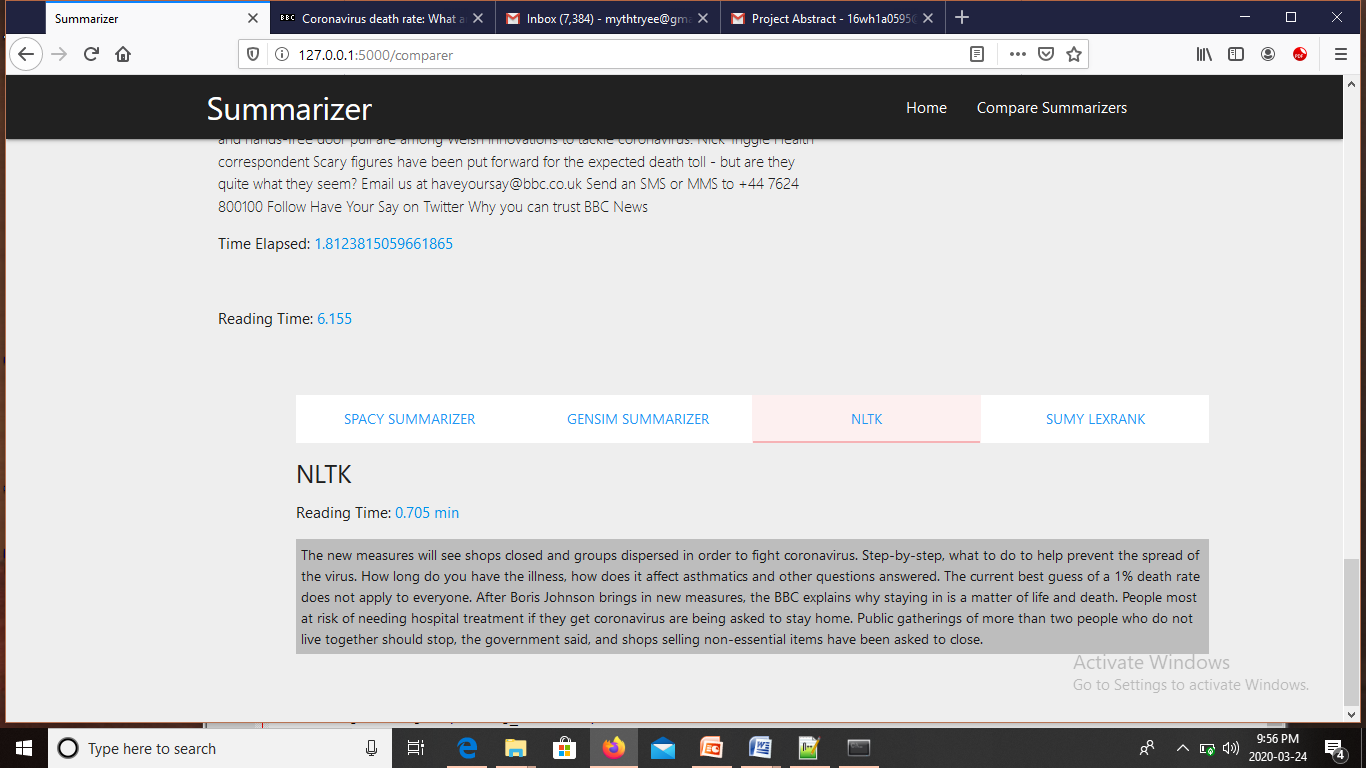
## Screenshot (49).png

**Fig 7:Uploading the News Article**

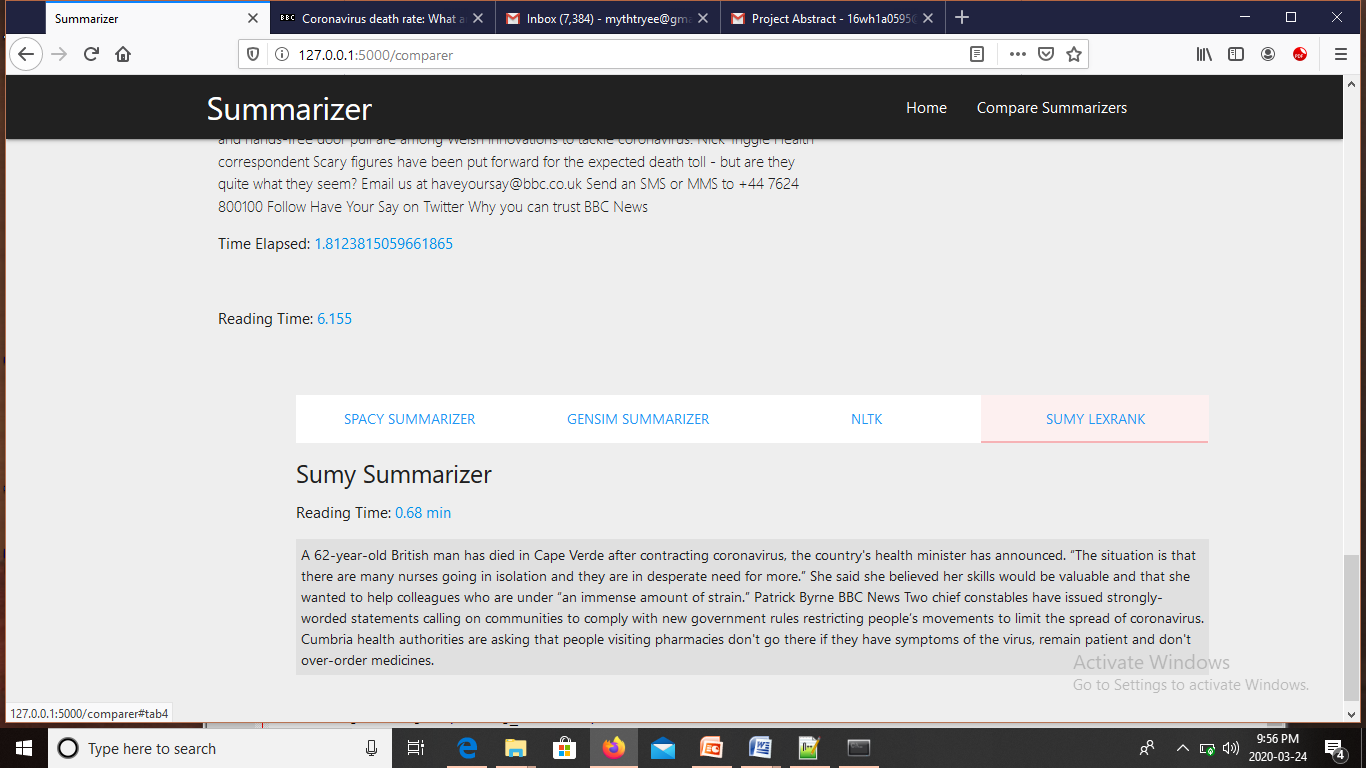
**4.5 Output Screenshots**

****

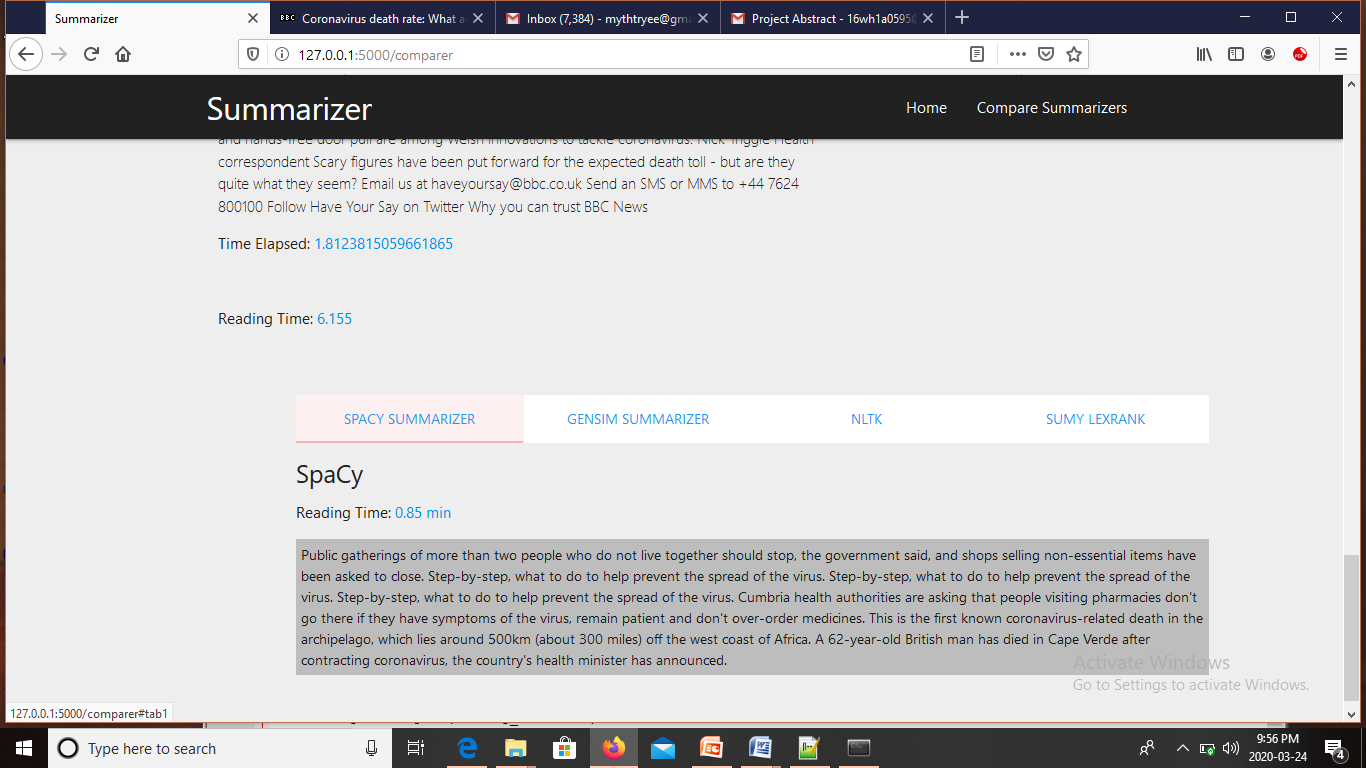
**Fig 8:Output Summary**

****

**Fig 9: Output summary using NLTK**

****

**Fig 10: Output summary using SUMY LEXRANK**

****

**Fig 11: Output summary using SpaCy**

**4.6 Result**

**Input article:**

Wi-fi web reaches farmers in Peru A network of community computer centres, linked by wireless technology, is providing a helping hand for poor farmers in Peru. The pilot scheme in the Huaral Valley, 80 kilometres north of the capital Lima, aims to offer the 6,000-strong community up-to-date information on agricultural market prices and trends. The Agricultural Information Project for Farmers of the Chancay-Huaral Valley also provides vital links between local organisations in charge of water irrigation, enabling them to coordinate their actions. More than 13,000 rural inhabitants, as well as 18,000 students in the region, will also benefit from the telecoms infrastructure. The 14 telecentres uses only free open source software and affordable computer equipment. The network has been three years in the making and was officially inaugurated in September. The non-government organisation, Cepes (Peruvian Centre for Social Studies) led the $200,000 project, also backed by local institutions, the Education and Agriculture ministries, and European development organisations. "The plan includes training on computers and internet skills for both operators and users of the system," said Carlos Saldarriaga, technical coordinator at Cepes. Farmers are also taking extra lessons on how to apply the new information to make the most of their plots of land. The Board of Irrigation Users which runs the computer centres, aims to make the network self-sustainable within three years, through the cash generated by using the telecentres as internet cafes. One of the key elements of the project is the Agricultural Information System, with its flagship huaral.org website. There, farmers can find the prices for local produce, as well as information on topics ranging from plague prevention to the latest farming techniques. The system also helps the inhabitants of the Chancay-Huaral Valley to organise their vital irrigation systems. "Water is the main element that unites them all. It is a precious element in Peru's coastal areas, because it is so scarce, and therefore it is necessary to have proper irrigation systems to make the most of it," Mr Saldarriaga told the BBC News website. The information network also allows farmers to look beyond their own region, and share experiences with other colleagues from the rest of Peru and even around the world. Cepes says the involvement of the farmers has been key in the project's success. "Throughout the last three years, the people have provided a vital thrust to the project; they feel it belongs to them," said Mr Saldarriaga. The community training sessions, attended by an equal number of men and women, have been the perfect showcase for their enthusiasm. "We have had an excellent response, mainly from young people. But we have also had a great feedback when we trained 40 or 50-year old women, who were seeing a computer for the first time in their lives." So far, the Huaral programme promoters say the experience has been very positive, and are already planning on spreading the model among other farmers' organisations in Peru. "This is a pilot project, and we have been very keen on its cloning potential in other places," underlined Mr Saldarriaga. The Cepes researcher recalls what happened in Cuyo, a 50-family community with no electricity, during the construction of the local telecentre site. There it was necessary to build a mini-hydraulic dam in order to generate 2kW worth of power for the computers, the communications equipment and the cabin lights. "It was already dark when the technicians realised they didn't have any light bulbs to test the generator, so they turned up to the local store to buy light bulbs," recalls Carlos Saldarriaga. "The logical answer was 'we don't sell any', so they had to wait until the next morning to do the testing." Now, with the wireless network, Cuyo as well as the other communities is no longer isolated.

**System Summary:**

We will definitely extend the maternity pay, from the six months where it now is to nine months, that's the extra £1,400." Ms Hewitt said: "We have already doubled the length of maternity pay, it was 13 weeks when we were elected, we have already taken it up to 26 weeks. The Tories dismissed the maternity pay plan as "desperate", while the Liberal Democrats said it was misdirected. Sandra Gidley said: "While mothers would welcome any extra maternity pay the Liberal Democrats feel this money is being misdirected." But David Frost, director general of the British Chambers of Commerce, warned that many small firms could be "crippled" by the move. She said new mothers were already entitled to 12 months leave, but that many women could not take it as only six of those months were paid. She said her party would boost maternity pay in the first six months to allow more women to stay at home in that time.

**Reference Summary:**

She said her party would boost maternity pay in the first six months to allow more women to stay at home in that time.She said new mothers were already entitled to 12 months leave, but that many women could not take it as only six of those months were paid.The Tories dismissed the maternity pay plan as "desperate", while the Liberal Democrats said it was misdirected.She said ministers would consult on other proposals that could see fathers being allowed to take some of their partner's maternity pay or leave period, or extending the rights of flexible working to carers or parents of older children.Liberal Democrat spokeswoman for women Sandra Gidley said: "While mothers would welcome any extra maternity pay the Liberal Democrats feel this money is being misdirected."We will definitely extend the maternity pay, from the six months where it now is to nine months, that's the extra £1,400."Ms Hewitt said: "We have already doubled the length of maternity pay, it was 13 weeks when we were elected, we have already taken it up to 26 weeks.Other plans include letting maternity pay be given to fathers and extending rights to parents of older children.

Rouge score : 0.68

**5.Conclusion and Future scope**

We were able to auto-summarize news articles and compare summaries generated by them to analyze what scoring parameters would lead to better results. In the process, we tweaked methods we had researched on to leverage the fact that we were dealing with news articles only. We found that journalists follow a fixed pattern to write a news article. They start with what happened and when it happened in the first paragraph and continue with an elaboration of what happened and why it happened in the following paragraphs. We wanted to use this knowledge while scoring the sentences by giving the nouns appearing in the first sentence a higher score. But after reviewing the preliminary results of our scoring method as described

Automatic text summarization is a complex task which contains many sub-tasks in it. Every subtask has an ability to get good quality summaries.

The important part in extractive text summarization is identifying necessary paragraphs from the given article. In this work we proposed extractive based text summarization by using statistical novel approach based on the sentences ranking the sentences are selected by the summarizer. The sentences which are extracted are produced as a summarized text.

**Future Enhancements:**

Current system generates an extractive summary of an article, we are only adding nouns in lexical chains. Adjectives too play a major role in defining the important sentences.

**The future work includes:**

* Adding adjectives also to the lexical chains along with nouns and then observe the effect on the summary generated.
* Exploring graph based algorithms for sentence scoring and extraction.
* Building RNN for abstractive text summarization.
* Evaluating the accuracy of generated summary by using bleu and rouge score.

**6.References**

[1] H. Dalianis, "SweSum – A Text Summarizer for Swedish," Technical report TRITA-NA-P0015,IPLab-174, NADA, KTH, October 2000.D.

[2] K. Sarkar, “An Approach to Summarizing Bengali News Documents,” in International Conference on Advances in Computing, Communications and Informatics, pp. 857–862, ACM, 2012.

[3] M. Kamal, “Text Summarization in Bioinformatics,” in 15th International Conference on Computer and Information Technology, pp. 592– 597, IEEE, 2012.

[4] Gupta,V and Lehal,G.s (2010). “A survey of text summarization extractive techniques.” Journal of Emerging Technologies in Web Intelligence,2(3),258-268.

[5] Nenkova, A.(2011).“Automatic summarization, Foundations and Trends in Information Retrieval”,5(2),103-233

[6] R. Jayashree, Srikanta Murthy K, and B. Anami, “Categorized Text Document Summarization in the Kannada Language by sentence ranking,” in 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 776–781, IEEE, nov 2012.

[7] Zhang and C.-h. Li, “Automatic text summarization based on sentences clustering and extraction,” in 2nd IEEE International Conference on Computer Science and Information Technology, vol. 1, pp. 167–170, IEEE, 2009.

[8] L. Yu and F. Ren, “A study on cross-language text summarization using supervised methods,” in International Conference on Natural Language Processing and Knowledge Engineering, pp. 1–7, IEEE, sep 2009.