NAME: BRIGHT SUMPUO SERINYE

INDEX NUMBER: 9413219

REFERENCE NUMBER: 20654382

COMPUTER SCIENCE 3

MINI PROJECT

TEXT SUMMARIZERIZATION APPLICATION

CHAPTER ONE: INTRODUCTION

1.1 INTRODUCTION

There is an ever-growing amount of data generated on a daily basis in this information age, therefore necessitating the development of a procedure to shorten long texts/documents immediately while keeping the main idea of it is necessary. Summarization also helps shorten the time needed for reading, fasten the search for information and help to get the most amount of information on one topic. The central object of computerized text summarization is decreasing the reference text into a smaller version while maintaining its knowledge alongside its meaning.

Text Summarization is the process of creating a summary of a certain document that contains the most important information of the original one, the purpose of it is to get a brief summary of the main points of the document. This is what Google models to some degree with the snippets shown in its search results. Abstractive summarization of multi-documents aims to generate a concentrated version of the document while keeping the main information. Due to the massive amount of data these days, the importance of summarization arose.

The summarization methods are grouped into abstractive and extractive summarization. An abstractive summary is an arbitrary text that describes the contexts of the source document. Abstractive summarization process consists of “understanding” the original text and “retelling” it in fewer words. An extractive summary, in contrast, is a selection of sentences (or phrases, paragraphs, etc.) from the original text, usually presented to the user in the same order i.e., a copy of the source text with most sentences omitted. An extractive summarization method only decides, for each sentence, whether or not it will be included in the summary. An abstractive summary can be further obtained from the extracted summary using techniques of sentence compression, theme intersection algorithms and also bring the cohesiveness in generated summary by using sentence ordering algorithms. The project aims to build abstractive summaries from the extractive summaries generated.

1.2 PROBLEM STATEMENT

To develop a model that does extractive text summarization along with finding and evaluating which parameters and machine learning and deep learning algorithms optimize the working of the model.

The scope of our project is to build a Deep Learning based solution that generates an Extractive and Abstractive Text summary of content available

To create a text summarizer which summarizes the text or the content of the paragraph in minimum words without changing its meaning. This system is made using NLP based model which is branch of machine learning. This text summarizer also summarizes text from the weblinks and also summarizes text from PDF document.

- Summaries reduce reading time.
- When researching documents, summaries make the selection process easier.
- Automatic summarization improves the effectiveness of indexing.
- Automatic summarization algorithms are less biased than human summarizers.
- Personalized summaries are useful in question-answering systems as they provide personalized information.
- Using automatic or semi-automatic summarization systems enables commercial abstract services to - increase the number of text documents they are able to process.

Research questions

Q1: To what degree does our summarized articles preserve the keywords from the original text?

Q2: How does the textual quality of the generated summaries differ from the original text?

Q3: How does Modified TextRank perform with regards to the ROUGE metric compared to other summarizers?

1.3 IMPORTANCE OF THE RESEARCH

Creating a summary from a given piece of content is a very abstract process that everyone participates in. Automating such a process can help parse through a lot of data and help humans better use their time to make crucial decisions. With the sheer volume of media out there, one can be very efficient by reducing the fluff around the most critical information. Due to the increasingly busy lifestyle of the people, they are not able to read their favorite, articles, documents, reviews, books, web articles, blogs, etc because doing these process is time consuming. Our motivation is to create a user friendly text summarization website to summarize web articles, passages and PDFs, which is free to use

1.4 SIGNIFICANCE OF PRIOR RESEARCH

Existing system

Existing system does not focus on Advanced NLP Techniques. End user does not get reliable summaries. Existing systems use Abstractive Text Summarization Technique, which does not give reliable summary. In this project, I focus on improving the existing system of the user.

Draw backs in existing system

i. summary is less accurate
ii. Time constraint is less
iii. Computational speed is more.
iv. It uses Abstractive Summarization.

Proposed system

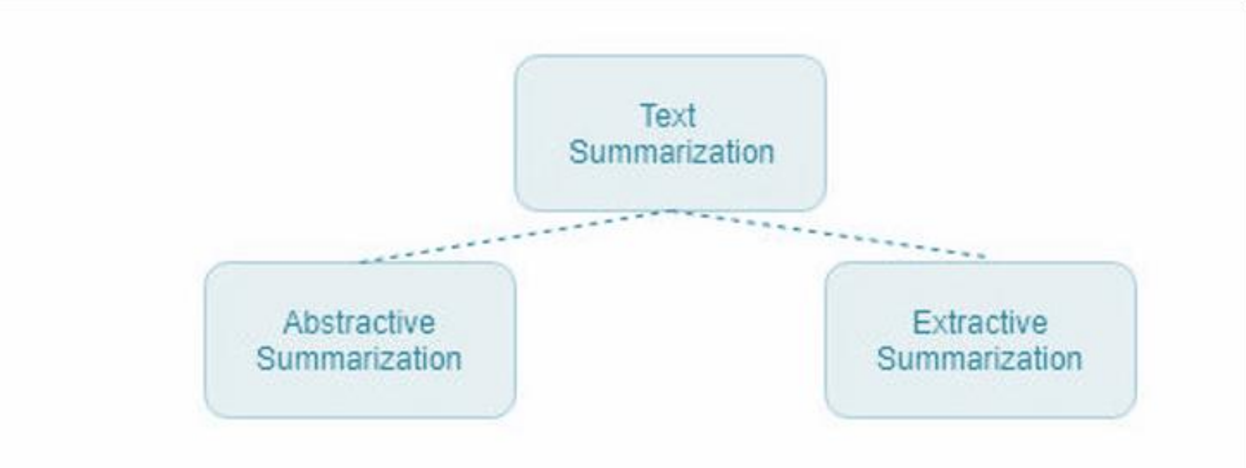
In the proposed system mainly focuses on providing a reliable summary. In the proposed system we are using Extractive Text Summarization which will provide reliable summary. An unsupervised Text Rank Algorithm is used for implementing Text Summarization. Advanced NLTK Techniques is used for making Graphical User Interface. This Project Focuses on providing a system which will provide an accurate Summary. In this Project, Input Can be Large Chunks of Text/web page urls and summarize the text

Advantages of proposed system

(i) My Algorithm executes with good performance because it is executing in a distributed environment.
(ii) Time Constraint is less
(iii) Computational Speed is more.
(iv) Accurate Summary
(v) It uses Extractive Summarization.

1.5 METHODOLOGIES

For obtaining text summarization, there are basically two major techniques i.e. Abstraction based Text Summarization and Extraction based Text Summarization.



An abstractive summarization tries to develop an understanding of the main concepts in a document and then express those concepts in clear natural language. It uses linguistic methods to study the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the salient information from the original text document. This method is the more difficult and it is poorly practical. It is highly complex as it needs extensive natural language processing.

An extractive summarization consists of selecting important sentences or paragraphs from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences. This method is fairly applicable and it usually gives reasonable result. Therefore research community is focusing more on extractive summaries, trying to achieve more coherent and meaning full summaries.

1.6 SCOPE

The project is wide in scope all of the limitations stated below may seem to contradict that, but they are the only restrictions applied. This project looks at single document summarization - the area of multi document summarization is not covered. Also, the summaries produced are largely extracts of the document being summarized, rather than newly generated abstracts. The parameters used are optimal for news articles, although that can be changed easily

WHAT DOES THE PROJECT NOT DO

It is equally important to know what the project cannot achieve and what should not be expected in the output of the system. The following points outline the same.

- It does not accept user input data corpus.

- It does not generate titles for the summaries generated.

- It does not summarize the image, graphs, videos or any other multimedia contents, which can be found in the news websites.

- It does not evaluate the summary using evaluation tools or algorithms. The evaluation is done manually based on predefined parameters.

- It does not let user view complete single news article, if he wishes to on portal. This can be viewed by looking in to json object if need be

1.7 ETHICAL AND SOCIAL ASPECTS

When training machine learning models on large amounts of text data, there is always a risk of inherent bias in the data. It is crucial to acknowledge this when deploying models for actual use since it could have negative implications if users are exposed to such biases. Furthermore, regarding news, factual correctness may also be an aspect to consider. Therefore, it is important to be wary of the sources of the data. However, humans are flawed, and one cannot completely rule out the possibility of some bias in this data. Text summarization systems are by no means perfect either, which is also important to consider when deploying them.

1.8 ANALYSIS

Typically, here is how using the extraction-based approach to summarize texts can work:

1. Introduce a method to extract the merited keyphrases from the source document. For example, you can use part-of-speech tagging, words sequences, or other linguistic patterns to identify the keyphrases.

2. Gather text documents with positively-labeled keyphrases. The keyphrases should be compatible to the stipulated extraction technique. To increase accuracy, you can also create negatively-labeled keyphrases.

3. Train a binary machine learning classifier to make the text summarization. Some of the features you can use include:

- Length of the keyphrase
- Frequency of the keyphrase
- The most recurring word in the keyphrase
- Number of characters in the keyphrase

4. Finally, in the test phrase, create all the keyphrase words and sentences and carry out classification for them.

Summary of a text is extracted following these techniques:

- Sentence Tokenization
- Word Tokenization
- Parts Of Speech Tagging
- Named Entity Recognition
- Stemming
- Punctuation Removal
- Text cleaning
- Word-frequency - Word’s frequency is calculated as - freq(word)/max(freq

1.9 CHAPTER SUMMARY

Nowadays, the need of automatic text summarization has augmented due to the rapid increase in number of information on the Internet. Therefore, it is too difficult for users to manually summarize those large online documents. text summarization solves this problem. It represents one of the natural language processing applications and is becoming more popular for information condensation. It allows getting the important information while dealing with large collection of documents. A good summary captures the essence of a long work in a brief informative statement that can be read and digested quickly. This solution can be developed using either extractive or abstractive approaches that both aimed at analyzing the texts and generalizing summaries. Text summarization by abstractive approach is stronger because it produces summary which is semantically related but difficult to generate. However, text summarization by extractive approach is easier for the human to program and for the computer to understand.

This project is focused on the using the concepts of extractive summary to summarize texts. The discussion revolves around the extractive approach Text summarization techniques can be applied helpfully depending on the user's needs.

CHAPTER TWO: LITERATURE REVIEW

This section deals with discussion of some of the existing techniques, a survey of work carried out by researchers in the domain of Text Summarization. This survey was carried out to know the existing techniques that are used for Text Summarization.

Document Clustering

Document Clustering, or Text Clustering, is a subfield of data clustering where a collection of documents are categorized into different subsets with respect to document similarity. Such clustering occurs without supervised information, i.e., no prior knowledge of the number of resulting subsets or the size of each subset is required.

Vasileios Hatzivassiloglou, Luis Gravano, Ankineedu MagantiAn investigate four hierarchical clustering methods - single-link, complete-link, groupwise-average, and single-pass.

Single-pass clustering makes irrevocable clustering assignments for a document as soon as the document is first inspected. Among the four techniques that they have considered, single-pass is then the best suited for the topic detection task, which requires systems to make clustering assignments “on-line” as soon as a new document is received. To investigate the limitations of such an on-line algorithm, they also experimentally compare the performance of single-pass with the other three clustering algorithms mentioned above. The other component of a clustering strategy that they explore in this paper is the document features that guide the clustering. Typically, document clustering techniques use the words that appear in the documents to define the “distance” function that determines the final clustering. But additional, more linguistically informed sets of features can be used in an attempt to limit the input features to the most important ones, facilitating the task of the learning (i.e., clustering) algorithm. In this paper they also investigate two such sets of automatically identified features: matched noun phrase heads, where additional pre-modifiers are excluded, and proper names (single nouns and phrases), categorized as people, place, or organizations' names.

Kathleen R. McKeown and her team at Columbia University describe the document clustering technique deployed in their product - *Newsblaster*. Newsblaster hierarchically classifies the news stories gathered by the crawler into three levels. At the top level, cosine similarity is used to compute the similarity between the news category and news article. At the next two lower levels, the system uses agglomerative clustering with a groupwise-average similarity function to group and identify similar events. It also incorporates a log-linear statistical model for automatically adjusting the relative weights of the different features.

Michael Steinbach George Karypis Vipin Kumar present the results of an experimental study of some common document clustering techniques in their paper. In particular, they compare the two main approaches to document clustering, agglomerative hierarchical clustering and K-means. Hierarchical clustering takes quadratic time whereas K-means takes linear time to perform clustering.

Extractive summarization

Extractive Summarization method consists of selecting important sentences, paragraphs etc. from the original document and concatenating them into shorter form. The importance of sentences is decided based on statistical and linguistic features of sentences.

Vishal Gupta discusses several methods of summarization used in past, namely Term Frequency-Inverse Document Frequency (TFIDF) method, Clustering based method, Graph theoretic approach, Machine Learning approach, LSA, An approach to concept-obtained text summarization, Text Summarization with Neural Networks, Automatic text summarization based on fuzzy logic and Text summarization using regression for estimating feature weights. The features that can be used for sentence selection and ordering are - Content word, title similarity, Noun phrase, title word, occurrence of non-essential terms such as “furthermore”, “additionally”, discourse based information, upper-case, biased words, pronouns, uppercase, sentence location, sentence length, sentence to centroid cohesion.

The paper also includes survey on NeATS. NeATS, a Multi-Document Summarizer, generates summaries in three stages: content selection, filtering, and presentation. The goal of content selection is to identify important concepts mentioned in a document collection. In a key step for locating important sentences, NeATS computes the likelihood ratio to identify key concepts in unigrams, bigrams, and trigrams, using the on topic document collection as the relevant set and the off-topic document collection as the irrelevant set. With the individual key concepts available, these concepts are clustered in order to identify major subtopics within the main topic. Clusters are formed through strict lexical connection. Each sentence in the document set is then ranked, using the key concept structures. NeATS uses three different filters: sentence position, stigma words, and maximum marginal relevance.

Dipanjan Das, Andre F.T. Martins, present a survey which intends to investigate some of the most relevant approaches both in the areas of single-document and multiple-document summarization, giving special emphasis to empirical methods and extractive techniques. Some promising approaches that concentrate on specific details of the summarization problem are also discussed. Special attention is devoted to automatic evaluation of summarization systems, as future research on summarization is strongly dependent on progress in this area. They describe in their paper various techniques used for generating summaries automatically, for each of the types - Single Document Summarization and Multi Document Summarization.

Under Single Document Summarization, they present a survey on execution and performance of - Machine Learning Methods, namely Naive Bayes, Rich Features and Decision Trees, Hidden Markov Models, Log-Linear Models, Neural Networks, - Deep Learning Language Methods. Under these they discuss two different discourse rhetorical structure - a binary tree representing relations between chunks of sentence and the one based on heuristics with the traditional features that have been used in the summarization literature. The discourse theory the Rhetorical Structure Theory (RST) that holds between two non-overlapping pieces of text spans: the nucleus and the satellite. Under Multi Document Summarization, they present a discussion on techniques namely Abstraction and Information Fusion, Topic Driven Summarization and MMR, Graph Spreading Activation and Centroid Based Summarization. They describe briefly some unconventional approaches that, rather than aiming to build full summarization systems, investigate some details that underlie the summarization process, and that we conjecture to have a role to play in future research on this field. These methods are - Short Summaries, Sentence Compression and Sequential Document Representation.

Abstractive Summarization

Abstractive Summarization consists of understanding the source text by using linguistic method to interpret and examine the text. The abstractive summarization aims to produce a generalized summary, conveying in information in a concise way, and usually requires advanced language generation and compression techniques. There are several techniques to generate Abstractive Summary. They could be obtained from the extracted sentences (extractive summary) or generated using Natural Language Generation techniques.

Atif Khan, Naomie Salim present a survey on abstractive text summarization methods in their paper. Abstractive summarization methods are classified into two categories i.e. structured based approach and semantic based approach. The main idea behind these methods has been discussed. Structured based techniques include - Tree Based, Template Based, Ontology Based, Rule Based. Semantic based techniques include - Multimodal Semantic, Information Item Based, Semantic Graph Based. Besides the main idea, the strengths and weaknesses of each method have also been highlighted. It is concluded from their literature studies that most of the abstractive summarization methods produces highly coherent, cohesive, information rich and less redundant summary. Jackie CK Cheung, in his thesis presents a distinction between abstractive and extractive summarization. The abstractive summarizer is the Summarizer of Evaluative

Arguments (SEA), adapted from GEA, a system for generating evaluative text tailored to the user's preferences. While experimenting with the SEA summarizer, they noticed that the document structuring of SEA summaries, which is adapted from GEA and is based on guidelines from argumentation theory, sometimes sounded unnatural. They found that controversially rated UDF features (roughly balanced positive and negative evaluations) were treated as contrasts to those which were non controversially rated (either mostly positive, or mostly negative evaluations). In SEA, contrast relations between features are realized by cue phrases, signaling contrast such as "however" and "although". These cue phrases appear to signal a contrast that is too strong for the relation between controversial and uncontroversial features. To solve this problem, they devise an alternative content structure for controversial corpora, in which all controversial features appear first, followed by all positively and negatively evaluated features.

Deep learning Deep

Learning has emerged as a new area of machine learning research. It tries to mimic the human brain, which is capable of processing and learning from the complex input data and solving different kinds of complicated tasks well. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text.

PadmaPriya, G. and K. Duraiswamy present an approach for multi document text summarization using Restricted Boltzmann Machine, a Deep Learning Technique. RBM is a graphical model for binary random variables. It consists of one layer of visible units (neurons) and one layer of hidden units. Units in each layer have no connections between them and are connected to all other units in other layer. They first perform pre-processing on the documents using techniques like part of speech tagging, stemming and stop-word removal. For feature computation they use four feature functions - title similarity, positional feature, term weight and concept feature. After obtaining a good set of feature vectors using RBM, they are further fine-tuned by using back propagation. Cross Entropy Error is used for adjusting the weight vectors. Two major problems faced while text summarization are ranking problem and selecting a subject of the top ranked sentences. They solve the ranking problem by finding the intersection between title and a particular sentence. They generate the sentence score for every sentence and these sentences are then arranged in the descending order based on the score. They obtain compression ratio as an input form the user. The no of top ranked sentences selected depends on this compression ratio. The evaluation matrices they consider are recall, precision and f-measure. They carry out the experiment for three different document set from different knowledge domain. This algorithm is sensitive to the input data. The proposed algorithm has satisfactory results.

Support Vectors for Clustering

Support Vector Machines (SVMs) provide a powerful method for classification (supervised learning). Use of SVMs for clustering (unsupervised learning) is now being considered in a number of different ways. Currently known research focuses on the field of image processing and simple data clustering.

Ben-Hur, Horn, Siegelmann and Vapnik present a novel clustering method using the approach of support vector machines. Data points are mapped by means of a Gaussian kernel to a high dimensional feature space, where they search for the minimal enclosing sphere. This sphere, when mapped back to data space, can separate into several components, each enclosing a separate cluster of points. They also present a simple algorithm for identifying these clusters. They further demonstrate the performance of their algorithm on several datasets. They also elaborately put forth the mathematics behind the technique and also essential formulae. They analyze the application of the algorithm by considering clustering with BSVs(Bounded Support Vectors) and without BSVs. They present examples suitably to demonstrate the same. Also they analyze the presence of overlapping and strongly overlapping clusters. They conclude the work, by stating that their algorithm has a distinct advantage over other, i.e. being based on a kernel method it avoids explicit calculations in the high dimensional feature space, and hence is more efficient. Also a unique advantage of their algorithm is that it can generate cluster boundaries of arbitrary shape, whereas other algorithms that use a geometric representation are most often limited to hyper-ellipsoids.

Kees Jong, Elena Marchiori, Aad van der Vaart introduce a heuristic method for non-parametric clustering that uses support vector classifiers for finding support vectors describing portions of clusters and uses a model selection criterion for joining these portions. Clustering is viewed as a two class classification problem and a soft-margin support vector classifier is used for separating clusters from other points suitably sampled in the data space. The method is tested on five real life datasets, including microarray gene expression data and array-CGH data. They briefly outline the algorithm by enumerating different stages. The SVC technique is not used so far for sentence clustering or document clustering.

Lexical Chains

A Lexical Chain is a sequence of related words in the text. It spans short(adjacent words or sentences) or long distances(entire text). It is independent of the grammatical structure of the text. It captures the cohesive structure of the text. Lexical chains are heavily used is the resolution of an ambiguous term and identification of the concept that the term represents. WordNet, part of speech tagging are mainly used in the identification of lexical chains. Since this results in the identification of the key section of the text, lexical chains can be used for Text Summarization. The summary generation using this technique relies on the model of topic progression in the text derived from lexical chains instead of its full semantic interpretation.

Regina Barzilay and Michael Elhadad present a novel approach to perform text summarization using Lexical Chains. They present a new algorithm to compute lexical chains in a text, merging several knowledge resources like WordNet, parts of speech tagger and shallow parser for identification of nominal groups and a segmentation algorithm. They carry out four steps in order to obtain the summary - segmenting the original text, constructing lexical chains, identifying the strong lexical chains and extracting the significant sentences. They present empirical result on the identification of strong chains among the possible candidates produces their algorithm. They describe how to identify significant sentences using lexical chains. They also present the preliminary evaluation of the results obtained by their method. These results indicate the strong potential of lexical chains as a knowledge source for sentence extraction.

Key-phrase Extraction

Extracting keywords is one of the most important tasks when working with text. Readers benefit from keywords because they can judge more quickly whether the text is worth reading. Alyona Medelyan describes that a typical keyword extraction algorithm has three main components:

- Candidate selection: Here, we extract all possible words, phrases, terms or concepts (depending on the task) that can potentially be keywords.
- Properties calculation: For each candidate, we need to calculate properties that indicate that it may be a keyword. For example, a candidate appearing in the title of a book is a likely keyword.
- Scoring and selecting keywords: All candidates can be scored by either combining the properties into a formula, or using a machine learning technique to determine probability of a candidate being a keyword. A score or probability threshold, or a limit on the number of keywords is then used to select the final set of keywords.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin and Craig G. Nevill-Manning describe the algorithm KEA (Keyword Extraction Algorithm) in their paper. Kea is an algorithm for automatically extracting keyphrases from text. Kea identifies candidate keyphrases using lexical methods, calculates feature values for each candidate, and uses a machine learning algorithm to predict which candidates are good keyphrases. The machine learning scheme first builds a prediction model using training documents with known keyphrases, and then uses the model to find keyphrases in new documents. They use a large test data corpus to evaluate Kea's effectiveness in terms of how many author-assigned keyphrases are correctly identified. G¨onen¸c Ercan [n] has tried to generate keyphrases by using lexical chains and machine learning algorithm - Naive Bayes Algorithm. In the thesis, he describes the feature functions that are used in the KEA algorithm. He has experimented with different feature functions that can be obtained from WordNet and lexical chains and suggested a few of them which yield best results. These are term frequency, first occurrence in text, last occurrence in text, semantic relation score, direct semantic relation score, lexical chain span, direct lexical chain span, hyponym and hypernym hierarchical levels, sentence count and direct sentence count.

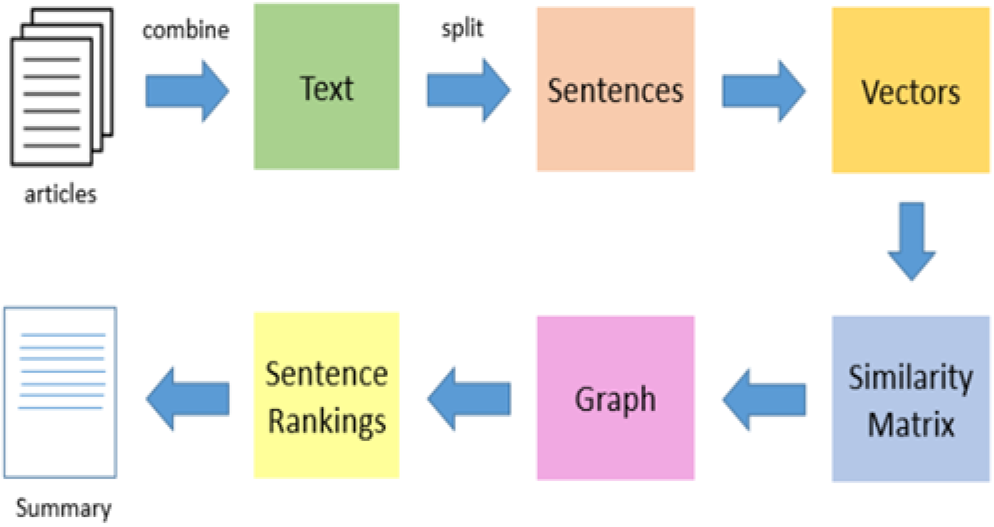
CHAPTER THREE: METHODOLOGY.

For the purposes of this project, I will be focusing on the use of the Extractive method working with the TextRank Algorithm

TextRank Algorithm

TextRank algorithm is a basic algorithm used in machine learning to summarized document. TextRank is an extractive and unsupervised text summarization technique. TextRank *does not rely on any previous training data* and can work with any arbitrary piece of text

- Similarity between any two sentences is used as an equivalent to the document transition probability
- The similarity scores are stored in a square matrix, similar to the matrix M used for PageRank. This is a similarity matrix
- First step is combine all the text in the documents to a document.
- Then, split this document into individual sentences.
- In the next step, we will find vector representation (words embedding) for each and every sentence
- We will be using cosine similarity to find similarity between sentences. Similarities between sentence vectors are then calculated and stored in a matrix.
- The similarity matrix is then converted into a graph, with sentences as vertices and similarity scores as edges, for sentence rank calculation.
- Finally, we will pink n sentences from the final summary



CHAPTER FOUR: IMPLEMENTATION AND TESTING

ALGORITHM SPECIFICATION

INPUT:- Large Chunks of Text/articles/WebPage Urls

OUTPUT:- Summarized Text and Audio File of Summarized Text

STEP1:- Concatenate all the contained in the articles

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

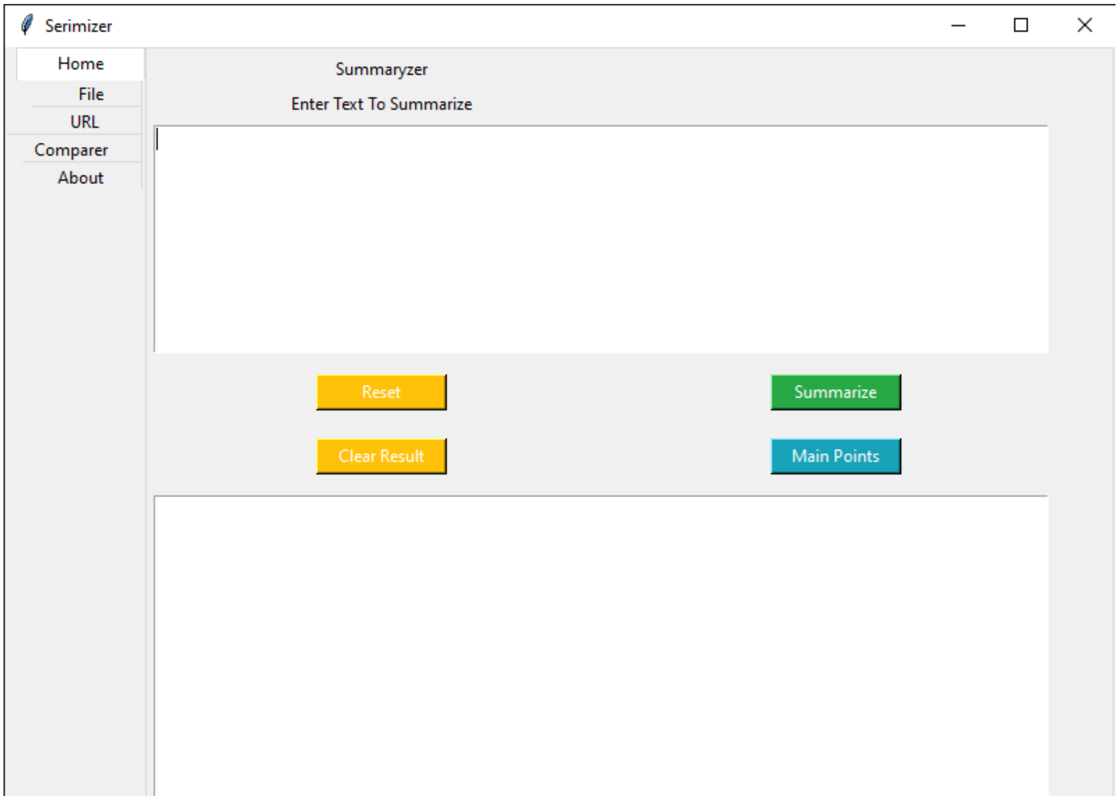
STEP3:- Find Vector Representation (Word Embeddings) for each and every sentence by using Glove Algorithm.

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

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

STEP6:- Find a certain number no top ranked sentences using page rank algorithm to form summary.

PROJECT IMPLEMENTATION



/*PYTHON LIBRARIES REQUIRED*/

```
# Core Packages
import tkinter as tk
from tkinter import *
from tkinter import ttk
from tkinter.scrolledtext import *
import tkinter.filedialog

# NLP Pkgs
from spacy_summarization import text_summarizer
from gensim.summarization import summarize
from nltk_summarization import nltk_summarizer

# Web Scraping Pkg
from bs4 import BeautifulSoup
from urllib.request import urlopen
```

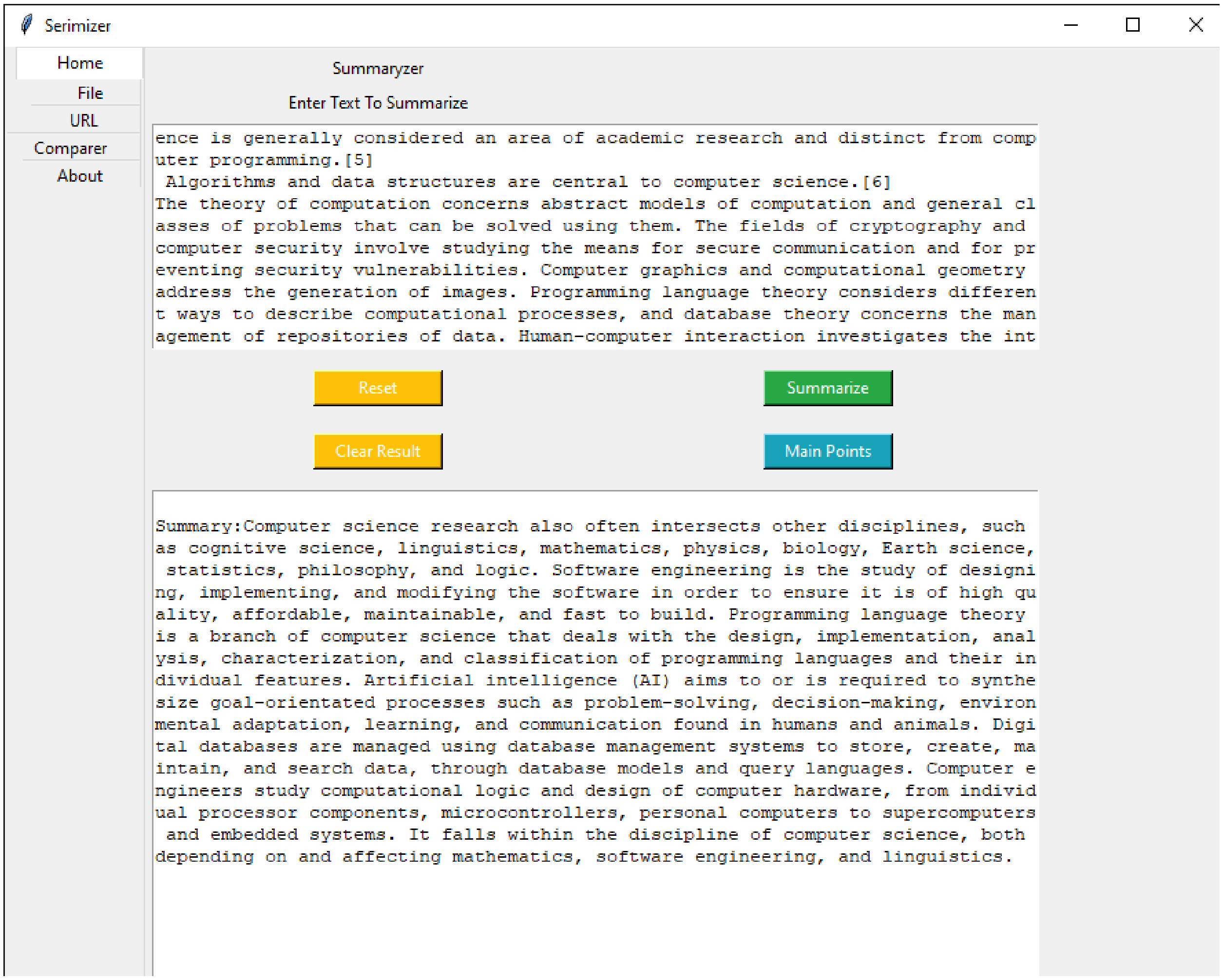
TESTING OF APPLICATION

Description of Test Scenario 1

(i) Copy the text which you want to summarize.

(ii) Click the summarize Button

(iii) Summary will be displayed



```
ence is generally considered an area of academic research and distinct from comp
uter programming.[5]
 Algorithms and data structures are central to computer science.[6]
The theory of computation concerns abstract models of computation and general cl
asses of problems that can be solved using them. The fields of cryptography and
computer security involve studying the means for secure communication and for pr
eventing security vulnerabilities. Computer graphics and computational geometry
address the generation of images. Programming language theory considers differen
t ways to describe computational processes, and database theory concerns the man
agement of repositories of data. Human-computer interaction investigates the int
```

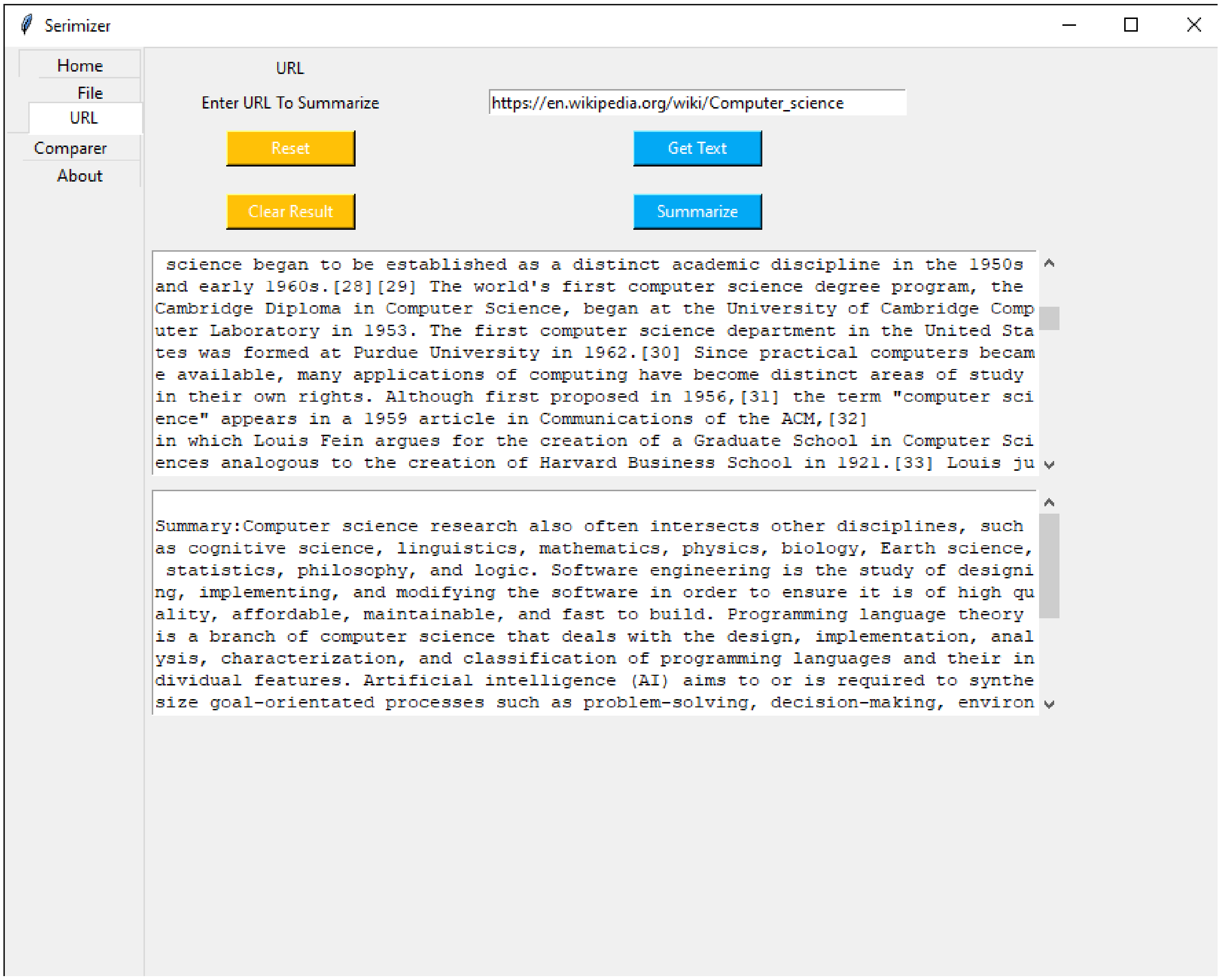
```
Summary:Computer science research also often intersects other disciplines, such
as cognitive science, linguistics, mathematics, physics, biology, Earth science,
 statistics, philosophy, and logic. Software engineering is the study of designi
ng, implementing, and modifying the software in order to ensure it is of high qu
ality, affordable, maintainable, and fast to build. Programming language theory
is a branch of computer science that deals with the design, implementation, anal
ysis, characterization, and classification of programming languages and their in
dividual features. Artificial intelligence (AI) aims to or is required to synthe
size goal-orientated processes such as problem-solving, decision-making, environ
mental adaptation, learning, and communication found in humans and animals. Digi
tal databases are managed using database management systems to store, create, ma
intain, and search data, through database models and query languages. Computer e
ngineers study computational logic and design of computer hardware, from individ
ual processor components, microcontrollers, personal computers to supercomputers
 and embedded systems. It falls within the discipline of computer science, both
depending on and affecting mathematics, software engineering, and linguistics.
```

DESCRIPTION OF TEST SCENARIO 2

Summarizing web page of computer science on Wikipedia

(i) Copy the web page url of “computer science”s on Wikipedia
(ii) Paste it in the URL bar of the text summarizer
(iii) Click “GET Text” button to get the text
(iv) Click “summarize” button
(v) Summary is displayed

S



```
 science began to be established as a distinct academic discipline in the 1950s
and early 1960s.[28][29] The world's first computer science degree program, the
Cambridge Diploma in Computer Science, began at the University of Cambridge Comp
uter Laboratory in 1953. The first computer science department in the United Sta
tes was formed at Purdue University in 1962.[30] Since practical computers becam
e available, many applications of computing have become distinct areas of study
in their own rights. Although first proposed in 1956,[31] the term "computer sci
ence" appears in a 1959 article in Communications of the ACM,[32]
in which Louis Fein argues for the creation of a Graduate School in Computer Sci
ences analogous to the creation of Harvard Business School in 1921.[33] Louis ju
```

```

Summary:Computer science research also often intersects other disciplines, such
as cognitive science, linguistics, mathematics, physics, biology, Earth science,
 statistics, philosophy, and logic. Software engineering is the study of designi
ng, implementing, and modifying the software in order to ensure it is of high qu
ality, affordable, maintainable, and fast to build. Programming language theory
is a branch of computer science that deals with the design, implementation, anal
ysis, characterization, and classification of programming languages and their in
dividual features. Artificial intelligence (AI) aims to or is required to synthe
size goal-orientated processes such as problem-solving, decision-making, environ
```

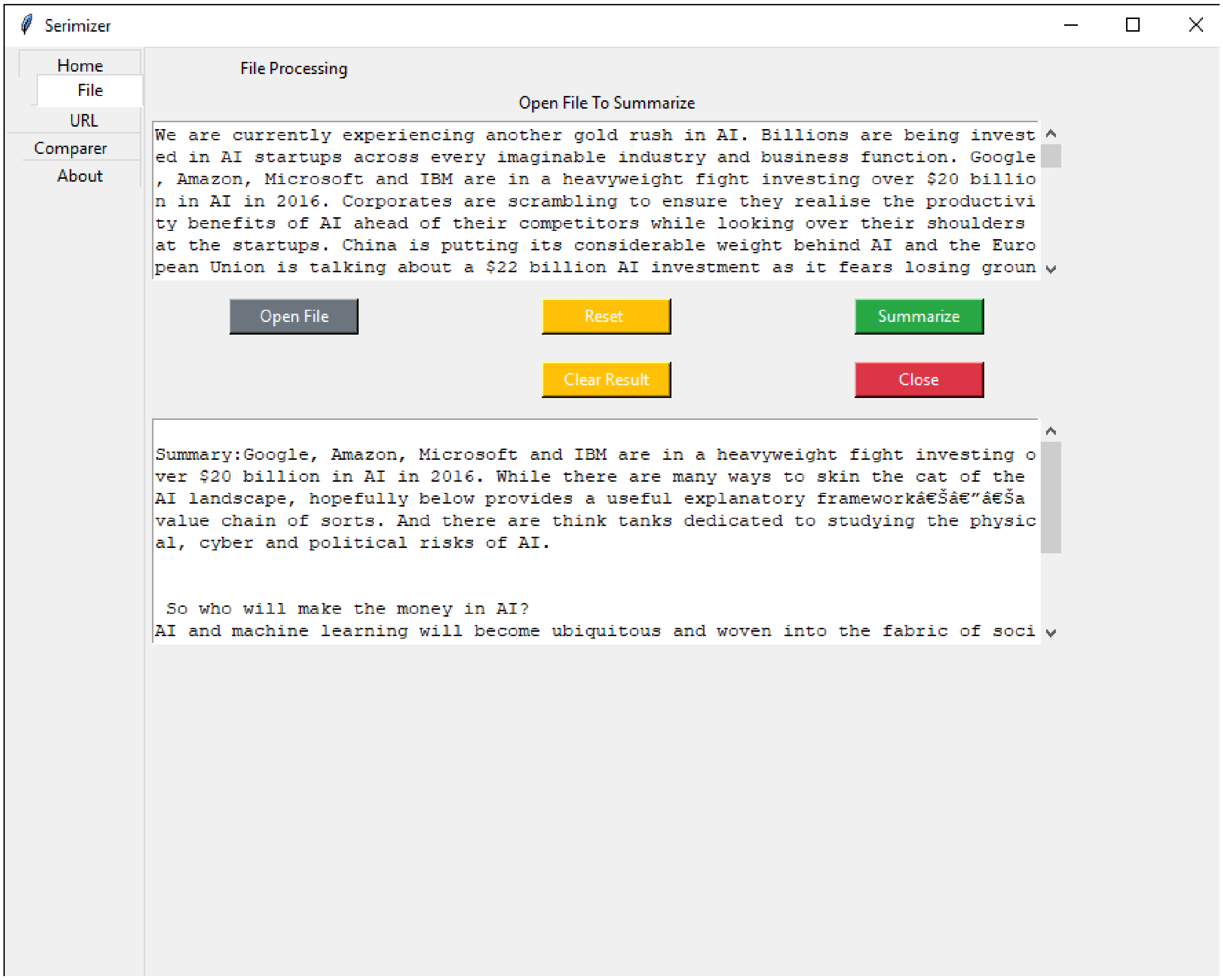
DESCRIPTION OF TEST SCENARIO3

(i) click “OPEN FILE” button

(ii) select text file from documents

(iii) text is read, then click the “Summarize” button

(iv) summary is generated



```
We are currently experiencing another gold rush in AI. Billions are being invest
ed in AI startups across every imaginable industry and business function. Google
, Amazon, Microsoft and IBM are in a heavyweight fight investing over $20 billio
n in AI in 2016. Corporates are scrambling to ensure they realise the productivi
ty benefits of AI ahead of their competitors while looking over their shoulders
at the startups. China is putting its considerable weight behind AI and the Euro
pean Union is talking about a $22 billion AI investment as it fears losing groun
```

```
Summary:Google, Amazon, Microsoft and IBM are in a heavyweight fight investing o
ver $20 billion in AI in 2016. While there are many ways to skin the cat of the
AI landscape, hopefully below provides a useful explanatory frameworkâ€Šâ€”â€Ša
value chain of sorts. And there are think tanks dedicated to studying the physic
al, cyber and political risks of AI.


 So who will make the money in AI?
AI and machine learning will become ubiquitous and woven into the fabric of soci
```

CHAPTER FIVE: SUMMARY, CONCLUSIONS AND RECOMMENDATIONS

Due to the tremendous increment of data on the web, extracting the most important data as a conceptual brief would be valuable for certain users. Because of the large volume of data deployed in digital space, there is a need to find a way to shorten texts and provide clear summaries. Summarizing the texts is still active in several research and needs further research and development in summarizing the texts Due to the huge increase in data available to us now, extracting the most important data as a conceptual summary will be useful to many users. In this project, I attempted to summarize large amounts of texts and explained the utility of short texts while preserving the original texts.

The domain of “Text Summarization” is quite huge and challenging by itself. Each component that make up to a final Automatic Text Summarizer is a research topic today. Hence, there is always a very good scope to enhance a summarizer system in terms of its capabilities and performance and add new and different dimensionalities to it.

That said, some of the future enhancements to the project, “Abstractive Multi-Document Text Summarization”, can be the following.

- The algorithms currently run on a corpus of size - 0. To enhance the usability and apply the project in real-time, different parameters used needs to be fine-tuned more. Hence, we need to run the algorithms on an even larger corpus.

- The document clustering algorithm can be further fine-tuned to improve the performance of the system. Also, a new algorithm or a different algorithm could be used, such that number of clusters of documents to be generated can be decided dynamically based on number of articles in corpus.

- More feature functions to generate feature vectors can be added. The feature functions can represent semantic aspects. The quality of feature functions determines the quality of output of algorithms. Hence more and better feature functions will enhance the performance of the system as a whole