Chapter 1

Introduction

1.1 Introduction

In the current era reading different documents and gathering all the required information in it is really difficult and the situation even worsens if there are multiple documents. User needs to read all of them and conclude the required details and after doing all this there is no guaranty that the user had studied the same document he really wanted to because there will be some cases where the headline is different and the topic inside it may be entirely different. Then the user needs to do all this work again and again until all the details required to the user are found. But this is a time taking process.

1.2 Text Mining and NLP

1.2.1 Text Mining:

Text mining, also referred to as text data mining, roughly equivalent to text analytics, is the process of deriving high-quality information from text. High-quality information is typically derived through the devising of patterns and trends through means such as statistical pattern learning. Text mining usually involves the process of structuring the input text (usually parsing, along with the addition of some derived linguistic features and the removal of others, and subsequent insertion into a database, deriving patterns within the structured data, and finally evaluation and interpretation of the output.

1.2.2 Natural Language Processing (NLP)

NLP is an area of computer science and artificial intelligence concerned with the interactions between computers and human (natural) languages, in particular how to program computers to fruitfully process large amounts of natural language data.

1.2.2.1 Stages in NLP

1.2.2.1.1 Lexical Analysis

Lexical Analysis involves identifying and the structure of words. Lexicon of a language means the collection of words and phrases in a language. Lexical analysis is dividing the whole chunk of txt into paragraphs, sentences, and words.

1.2.2.1.2 Syntactic Analysis (Parsing)

Syntactic Analysis involves analysis of words in the sentence for grammar and arranging words in a manner that shows the relationship among the words. The sentence such as "The school goes to boy" is rejected by English syntactic analyser.

1.2.2.1.3 Semantic Analysis

Semantic Analysis draws the exact meaning or the dictionary meaning from the text. The text is checked for meaningfulness. It is done by mapping syntactic structures and objects in the task domain. The semantic analyser disregards sentence such as "hot ice-cream"

1.2.2.1.4 Discourse Integration

The meaning of any sentence depends upon the meaning of the sentence just before it. In addition, it also brings about the meaning of immediately succeeding sentence. So in Discourse Integration gives the meaning based on all the sentences given before it.

Eg. Consider the sentence "Water is flowing on the bank of the river"

But bank has two meanings One Financial Institute and Two River of the bank here System has to consider the second meaning.

1.2.2.1.5 Pragmatic Analysis

During this, what was said is re-interpreted on what it actually meant. It involves deriving those aspects of language which require real world knowledge.

1.3 Information Extraction:

Information Extraction is a task of automatically extracting structured information from the given raw data. In many cases this activity uses Natural Language Processing. This is the ba-

sis for Single Document Summarization. Information extraction has started in the end of 1970's and JASPER built by REUTERS was the first commercial product built using this with an aim of providing real time financial news to financial traders.

1.3.1 Approaches to Information Extraction:

1.3.1.1 Ontology based Information Extraction (Daya C. Wimalasuriya, Mar 2010)

1.3.1.2 Using Classifiers like Naive Bayes classification etc…

1.3.1.3 Sequence models like Conditional random fields (Peng, F.; McCallum, A. 2006; Shimizu, Nobuyuki; Hass, Andrew 2006), hidden Markov's model, Maximum Entropy Markov Model.

1.3.2 Sub Tasks in Information Extraction:

1.3.2.1 Pre-Processing of the text

This is done using the computational tools such as tokenization(Splitting the sentences into tokens), Sentence splitting(Splitting the entire document given into sentences), Stemming (Finding the root word for all the different words like running, runs, run, will run all will be stemmed to a root word "RUN"), Morphological Analysis etc…

1.3.2.2 Getting rid of the noise

Noise means a sudden extremity in the data which affects the entire data thereby reducing the efficiency of the given data. So this should be removed to get a higher efficiency. To eliminate noise developers follow a wide range of techniques like smoothening, binning etc….

1.3.2.3 Connecting the concepts

Connecting the concepts is the task of identifying relationships between the extracted concepts. There will be some problems like Entity identification. Eg: customer_id in one table and cust_id are two tables are same.

1.3.2.4 Unifying

This is about presenting the extracted data into a standard form.

1.4 Text Summarization:

Text Summarization is the process of shortening a text document with software, in order to create a summary with the major points of the original document. Technologies that can make a coherent summary take into account variables such as length, writing style and syntax.

It is a hybrid between Feature Extraction, Statistical Analysis and Data Mining. The most important idea of summarization is to find the subset of data which contains the information of the entire set. Search Engines are the best examples of Summarization.

1.4.1 Stages of Text Summarization:

There are mainly three stages for summarization. They are

1.4.1.2 Topic Identification:

Topic identification is the first and primary stage in the text summarization. This is the most important step because only based on this user can get some key words and give importance to the topic based on this and give priority to the topic related words such as jargons etc…

1.4.1.3 Interpretation:

Interpretation often occurs when reading or even writing literature papers or articles etc…. The system should interpret the meaning of the literary work or a document or anything else that it reads to generate a summary from the given documents but simply stops there. Instead of trying to analyse the deeper meaning of the work, which is what analysing is, the system interprets what the document portrays.

1.4.1.4 Summary generation:

Summary generation is the major step in document summarization. After interpreting what the document is, the system then moves to this stage where the entire document is read once again and all the important points are generated as a summary so that the user can read the summary and get a glance at the entire document. If the user wants more information then the user can read the entire document.

1.4.2 Methods in Text Summarization:

There are two general approaches to automatic summarization:

1.4.2.1 Extractive Text Summarization:

Extractive methods work by selecting a subset of existing words, phrases, or sentences in the original text to form the summary. This gives the output which has reduced number of sentences gives only the most important sentences using sentence scores etc.... But either the words or the grammar of the sentences do not change.

1.4.2.2 Abstractive Text Summarization:

Abstractive Methods build an internal semantic representation and then use natural language generation techniques to create a summary that is closer to what a human may think. Such a summary includes verbal changes i.e., there will be change in the words present in the summary and the given documents.

1.5 Steps in Text Pre-processing:

1.5.1 Text Normalization:

Text normalization is a process of transforming text into a single canonical form. Normalizing text before storing or processing it allows for separation of required data from the rest so that the system can send consistent data as an input to the other steps of the algorithm.

1.5.2 Stop Word Removal

Stop Word: A Stop Word is a commonly used word in any natural language such as “a, an, the, for, is, was, which, are, were, from, do, with, and, so, very, that, this, no, yourselves etc….”.(Appendix A)

These Stop Words will have a very high frequency and so these should be eliminated while calculating the term frequency so that the other important things are given priority. Stop word removal is such a Pre-processing step which removes these stop words and thereby helping in the further steps and also reducing some processing time because the size of the document decreases tremendously.

Consider a Sentence

"This is a sample sentence, showing off the stop word removal".

Output after Stop word removal is:

["sample", "sentence", "showing", "stop", "word", "removal"]

Note: Though Stop words refer to the most commonly used words in a particular language, there is no single universal list of stop words, different tools uses different stop words.

1.5.3 Stemming:

Stemming is a pre-processing step in Text Mining applications as well as a very common requirement of Natural Language processing functions. In fact it is very important in most of the Information Retrieval systems. The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

Eg: A stemmer for English should identify the strings "cats", "catlike", "catty" as based on the root "cat".

1.5.3.1 Rules of Suffix Stripping Stemmers:

1.If the word ends in 'ed', remove the 'ed'.

2.If the word ends in 'ing', remove the 'ing'.

3.If the word ends in 'ly', remove the 'ly'.

1.5.3.2 Rules of Suffix Substitution Stemmers:

1.If the word ends in 'ies' substitute 'ies' with 'y'.

Generally this stemmer is used because of some word like families etc…

1.6 Classification:

Text Summarization can be done by using either a single document or by using more than one documents.

1.6.1 Single Document Summarization:

This kind of summarization takes a single document as an input and summarizes the same. There will be a limited usage in the real time because, one can't derive the entire information from a single document. Inorder to get the information from more than one documents the user have to do the single document summarization for each and every document. This is a monotonous thing and no user will be happy doing the same thing again and again.

1.6.2 Multiple Document Summarization:

In Multiple Document Summarization the user can select multiple relevant documents and can directly get the summary of all the documents. In this kind redundancy can also be reduced thereby increasing the efficiency. Also user can spend less time on reading the summary of different documents.

1.7 Evaluation Measures:

1.7.1 Precision:

Precision is well-suited to evaluating problems where the goal is to find a set of items from a larger set of items. In NLP, this can correspond to finding certain linguistic phenomena in a corpus.

Precision represents the proportion of items or entities that the system returns which are accurately correct. It rewards careful selection, and punishes over-zealous systems that return too many results: to achieve high precision, one should discard anything that might not be correct. False positives – spurious entities – decrease precision.

$$P = \frac{|\text{truepositives}|}{(|\text{truepositives}|+|\text{falsepositives}|)} \qquad \text{------ (1.1)}$$

1.7.2 Recall:

Recall is another important evaluating measure to express the goodness of the Natural Language Processing System.

Recall indicates how much of all items that should have been found, were found. This metric rewards comprehensiveness: to achieve high recall, it is better to include entities that one

is uncertain about. False negatives – missed entities – lead to low recall. It balances out precision.

$$R = \frac{|\text{TruePositives}|}{(|\text{TruePositives}|+|\text{falsenegatives}|)} \qquad \text{-------- (1.2)}$$

	p' (Predicted)	n' (Predicted)
p (Actual)	True Positive	False Negative
n (Actual)	False Positive	True Negative

Table 1.1 Measuring the correctness of the system

p: Originally the statement is True

n: Originally the statement is False

p': System has predicted the statement as True

n': System has predicted the statement as False

From these above, every outcome of analysis of a review will have 4 states:

1. True Positive: Originally the statement is True and the System has predicted the statement as True. So the System is correct.

2. False Positive: Originally the statement is False But, the System has predicted the statement as True, so the system was wrong.

3. False Negative: Originally the statement is True But, the System has predicted the statement as False, so the system was wrong.

4. True Negative: Originally the statement is False and the System has predicted the statement as True, so the system was correct.

1.8 Motivation for the work:

Now-a-days, time is the most important thing in daily life. No-one is ready to read an entire document and then think whether the information is relevant or not. So there should be a tool to generate the summary containing all the required details of the text. At the same time there is a high probability almost 100% to find different article present on the same topic. This worsens the situation. The user needs to study all the documents which sometimes may take some hours to study and understand all the documents.

1.9 Problem Statement:

In this current era, reading an entire article and gathering required information from it is itself difficult and along with it the user sometimes find more than one article on the same topic. To gather all the important details from it becomes much more difficult. So, a user feels better if he has some tool to identify all the important points present in the documents, because in the current modern era everyone wants to find a shortcut of doing things so that they can save time which they can utilize to do some other important thing.

This project is done on the basis of the above problem which takes multiple text documents as an input and gives the most important details present in them basing on the compression ratio given by the user. This project is mainly divided into two phases. In the first phase it finds the sentence scores of each sentence in all the multiple documents. In the second phase it selects the sentences with highest scores in both the documents as an output.

1.10 Organization of the Project Report:

Remaining chapters of the report are described as follows

Chapter 2 describes about the literature survey, Summaries of different papers studied, different methods that can be used for Text Summarization

Chapter 3 describes about the methodology, pre-processing steps and algorithm implemented for the project, scope and feature of proposed method.

Chapter 4 contains the implementation code, system configuration and screenshots of the output.

Chapter 5 contains the conclusion and scope for further studies in the project.

CHAPTER 2

LITERATURE SURVEY

2.1 INTRODUCTION

The work focuses on generating a multiple text document summarization using feature extraction and TF-IDF. In this new era, where tremendous information is available on the Internet, it is most important to provide the improved mechanism to extract the information quickly and most efficiently. It is very difficult for human beings to manually extract the summary of large documents of text. There is plenty of text material available on the Internet. So there is a problem of searching for relevant documents from the number of documents available, and absorbing relevant information from it. In order to solve the above two problems, the automatic text summarization is very much necessary. Text summarization is the process of identifying the most important meaningful information in a document or set of related documents and compressing them into a shorter version preserving its overall meanings. Before going to the Text summarization, firstly, have to know that what a summary is. A summary is a text that is produced from one or more texts, that conveys important information in the original text, and it is of a shorter form. The goal of automatic text summarization is presenting the source text into a shorter version with semantics. The most important advantage of using a summary is, it reduces the reading time of the user so that the user can utilize this time on analyzing things.

2.2 METHODS FOR GENERATING DOCUMENT SUMMARISATION

2.2.1 A Review on Data Merging Techniques & Multi-Document Summarization

Their research to present an investigation of existing methods with the interests highlighting the need of clever Multi-Document summarizer. Jyoti Singh et al (2017).

In this it tells about different approaches like:

1. Cluster Based Approach
2. LDA method
3. Ranking Based Approach

Cluster Based Approach:

Gathering technique in a general sense incorporates only three undertaking as pre-dealing with, grouping and outline time. The going with procedure must be done before offering commitment to the gathering strategy by using pre-get ready. Basically, predealing with steps isolated into taking after core interests.

- Tokenization: It breaks the substance into detached lexical words that are confined by white space, comma, dash, spot et cetera. Stop words ejection: Stop words like an, about, all, et cetera., or other zone subordinate words that must be removed.
- Stemming: It clears increases like "s", "ing" in this manner on from documents.
- Precision: It is defined as the fraction of retrieved docs that are relevant given as Relevant = P (relevant | retrieved)

$$Pn = \frac{\text{m}}{N-n+1} \qquad \text{------(2.1)}$$

where,

n =retrieved documents

m = relevant documents

N = no. of documents

- Recall: Fraction of relevant docs that are retrieved given as Retrieved = P (retrieved | relevant)

$$Rn = \frac{m}{n} \qquad \text{-------(2.2)}$$

- TF-IDF:

$$TF(term, document) = \frac{\text{Frequencyofterm}}{\text{No.OfWordsinDocument}} \qquad \text{------(2.3)}$$

IDF (inverse document frequency): It calculates whether the word is rare or common in all documents. IDF (term, document) is obtained by dividing total number of Documents by the number of documents containing that term and taking log of that.

$$IDF(term, document) = \log\left(TotalNo.of\frac{doc}{No}.ofdoccontainingterm\right) \quad \text{----(2.4)}$$

TF-IDF: It is the multiple of the value of TF and IDF for a particular word. The value of TF-IDF increases with the number of occurrences within a doc and with rarity of the term across the corpus.

$$\text{TF-IDF=TF*IDF} \quad \text{-----(2.5)}$$

This paper is using K Means Clustering Algorithm addresses K-MEAN count. K-mean bundling is used to social affair all the relative course of action of records together and segment the report into k-assemble where to find k centroids for each cluster. (Harshal J. Jain et al. , 2012)

This paper is on Word Sequence Models for Single Text Summarization. It is the Extractive abstract methodology which gives a blueprint to the customer for practically identical substance documents. (Rene Arnulfo Garcia-Hernandez et al. 2009)

In this work, n-gram is used as a segment of a sentence in an unsupervised learning strategy. This methodology is used for clustering the similar sentences and structures the gatherings where most illustrative sentences are chosen for delivering the layout. The count portrayed as takes after-

- Pre-Processing: First, remove the stop words, evaluate clamor and afterward apply stemming process on it.
- Term choice must be taken what size of n-grams as highlight is to be utilized to speak to the sentences.
- Term weighting: choice must be taken that how every components are aligned.
- Sentence grouping: choose the contribution for the k-mean calculation.
- Sentence choice: after completing k-mean calculation; pick the closest sentence to every centroid for producing the outline. It gives a rundown to the client for comparative con-

tent reports. It is important to discover from the earlier method for deciding the best gram estimate for content synopsis what is not clear how to do.

Latent Dirichlet Allocation (LDA) method :

LDA is scheduled into three errands: First, scattering of topic is done over the subject which is tried from a Dirichlet transport. Second, a singular point is picked by this apportionment for each word in the record. Finally, every word is tried from a polynomial scattering over words which are described in investigated point. In addition, get the title information and the substance information in reasonable way which is valuable in execution of Summarization.

As multi-record rundown covers unmistakable events from the sentences in the reports and LDA isolate that chronicles into different subjects or events. However, here orthogonal vector is required to reduce consistent information substance and it gives relationship of sentences. SVM is used to get the orthogonal portrayals of vectors and moreover can addresses as sentence orthogonal. LDA finds assorted subjects in the documents however SVD finds the sentences which are best address these focuses. This paper Multi-document Summarization in perspective of Hierarchical Topic Model addresses h-LDA (different leveled Latent Dirichlet Allocation) figureuring exhibited for extractive multi-report outline strategy. h-LDA computation disconnect into four phases as Pre-get ready of the instructive list, Sentence weighting, Similarity Calculation and Summary sentence weight. It addresses productive probabilistic model. This concentrates sit without moving topics from different records and besides can mastermind these subjects into a pecking request to increment semantic examination. Meanwhile sentence weight advancement is used to correct the diagrams. So by doing this, user get brief rundown. Here TAC 2010 datasets are used for exploratory reason and besides ROUGE technique is used for evaluating the results. It gives favored results over standard system. This computation isolates the topic into two classes as important subject and insignificant point. Colossal point as LDA character of sentence technique is used as a piece of this proposed show for checking similarity between sentence topic. This approach highlights the benefits of bits of knowledge characteristics and teamed up with LDA point show. LDA highlight is used to discover sentence weight. (HongyanLill et al. , 2011)

Ranking Based Approach:

Firstly, name the sentences and get the semantic parts, and thereafter apply a novel SR-Rank computation. SR-Rank estimation in the meantime positions the sentences and semantic parts; it expels the most fundamental sentences from a file. An outline based SR-Rank count rank all sentences center points with the help of various sorts of center points in the heterogeneous diagram.

2.2.2 Towards Efficient model for Automatic Text Summarization

This paper presents the current technologies and techniques as well as prevailing challenges in automatic text summarization, consequently, this is a proposed model for improving text summarization by using a method that combines sentence scoring algorithm with sentence reduction.

The method used for automatic text summarization can either be extractive or abstractive. Extractive summarization method involves picking important sentences from a document while abstractive method of summarization involves the use of linguistic methods to analyze and interpret a document, the system then looks for another way to portray the content of the document in a short form and still pass across the main gist of the document. Also the input of a text summarization system can either be single or multiple. Single document summarization involves summarizing a single text while Multi-document summarization involves summarizing from more than one source text.

2.2.2.1 Sentence Scoring And Sentence Reduction Models:

Sentence score is a value that determines the sentences that are relevant to the input text. The input to the system is a single document.Sentence scoring occurs at the first stage; significant sentences are identified and extracted. The second stage involves the sentence reduction module; the extracted sentences from the sentence scoring module are processed, grammar checking and removal of target structures is done.

Sentence Scoring Module:

Sentence scoring module In the sentence scoring module, there are two major steps involved:

1. Preprocessing:

This step involves the removal of stop-word and tokenization; stop-words are extremely common words (e.g. a, the, for). For this part, a stoplist which is a list of stop-words is used. Tokenization involves breaking the input document into sentences.

2. Sentence scoring:

After the document has been broken into group of sentences. Sentences are extracted based on three important features; sentence resemblance to query, cue phrases and word frequency.

3. Sentence resemblance to query:

This is modelled after sentence resemblance to title which calculates a score based on the similarity between a sentence and the title of a document. So sentence resemblance to query calculates a score based on the similarity between a sentence and the user query which means that any sentence that is similar to the query or it includes words in the query are considered important.

And the score will be calculated using the following formula:

$$Score = \frac{No.of\,query\,words\,in\,Sentence}{\text{nQW}} \qquad \text{------(2.6)}$$

where

nQW = number of words in query

4. Cue Phrases:

The justification of using this feature is that the presence of some words likes “significantly”, “Since” point to important gist in a document and a score is assigned to such sentences. The score is computed using

$$Score = \frac{\text{CuePhraseCout}}{\text{No.ofCuesinSentence}} \qquad \text{------(2.7)}$$

5. Word frequency:

It is a useful measurement of significance because it is revealed in that an author tend to repeat certain words when trying to get a point across. So sentences that contain frequently occurring words are considered to be significant.

Sentence Reduction Module:

In the sentence scoring module, the original document and the extracted sentences from the sentence scoring module is processed so as to remove irrelevant phrases from the document to make the summary concise, the sentence reduction algorithm is described in detail. The processing involves: Syntactic Parsing Stanford parser, a syntactic parser is used to analyze the structure of the sentences in the document and a sentence parse tree is produced. The other stages involved in the sentence reduction module add more information to the parse tree, these information aids the final decision to be made. Sentence Correction,System goes back and forth on the sentence parse tree, node by node to identify parts of the sentence are important and must not be removed to make the sentence grammatically correct. For example, in a sentence, the main verb, the subject, and the object(s) are essential if they exist. Removal of Target Structures For this research work will be using the main clause algorithm for sentence reduction. In this algorithm, the main clause of a sentence is obtained and in that main clause user identifies the target structures which are the structures to be removed and they are adjectives, adverbs, appositions, parenthetical phrase, and relative clauses. A reduced sentence is gotten after the targeted structures has been identified and removed from the sentence parse tree. Summary Construction, user once again goes back and forth on the sentence parse tree and see if the reduced sentences are grammatically correct. The reduced sentences are then merged together to give the final summary. After the sentence reduction module carries out all four steps, a concise and coherent summary is expected as output

2.2.3 Literature Review of Automatic Multiple Documents Text Summarization

In this paper, concepts of multiple documents text summarization are reviewed that categorize different approaches in this ground.

This literature review explore the recent trends in summarization system that comes from novice procedure to this time of computer, where natural language processing is used to generate the

summary resemble with human expert. Almost all the techniques found for summarization presumed that the documents of correlated topic will be submitted for abstraction. (Fukumotoj., 2014)

2.2.3.1 A Review On Automatic Multiple Documents Text Summarisation:

2.2.3.1.1 Rank Based Approach:

The algorithm used here applies a spreading activation technique to discover nodes related to the core theme. Consecutively the method finds neighbour of starting nodes and accumulate the activating nodes to the output.

A vertex is added for each sentence and link between vertices are set up using sentence similarity relation. Then top scored sentences are chosen to construct abstract. A new approach under the hub-authority framework has been introduced here that unites the text content with some cues and investigates the sub-topics into graphbased sentence ranking algorithm for generating expected output. This is a two-link graph including both sentences and documents. It is assumed that the sentences which belong to an important document, highly correlated with the document, will be more likely to be chosen into summary. (InderjeetMani, 1997), (Junlin Zhang, 2005), (Xiaojun Wan, 2008),(Rada Mihalcea, 2004)

2.2.3.1.2 Time Based Approach:

Kathleen McKeown, 1995, This method focused on techniques to summarize how the trends of an event changes over time, using various points of view over the same event or series of events.

Here the enhancement of TextRank is unveiled named TimedTextRank with incorporating time dimension. This is based on the proclamation that for an evolving topic, recent documents are usually more important than earlier documents. (Xiaojun Wan, 2007)

2.2.3.1.3 Sentence Co-releation Based Approach:

A link between two sentences is considered as a vote cast from one sentence to the other sentence. Sentences will be extracted based on casted votes, scores, position etc. to get the abstract. This research emphasized on logical-closeness rather than topical-closeness which is

based on synonymy and not strong enough to measure the coherence of sentences. (ShanmugasundaramHariharan, 2012), (Tiedan Zhu, 2012)

Clustering Based Method:

Using clustering, coverage, anti redundancy and summery cohesion criteria the proposed procedure emphasized on "relevant novelty" which is a metric for minimizing redundancy and maximizing both relevance and diversity. This method combines Clustering, Linguistics and Statistics for Summarization and named it CLASSY. Structural design made up of five steps: preparation of raw texts, trimming of sentences, scoring, redundancy elimination and sentence organizing. Two models were proposed here in the process of sentence ranking. One is to incorporate the cluster-level information into the link graph. Second is to consider the clusters and sentences as hubs and authorities in the HITS algorithm for scoring sentences. This technique generates a summary in a query-oriented fashion with an unsupervised approach called SciSumm. Here the proposed method has four principal modules: text tilling, clustering, ranking and summery presentation. (Jade Goldstein, 2000), Judith D. Schlesinger, 2008), (Xiaojun Wan, 2008), (Nitin Agarwal, 2011)

2.2.3.1.5 Term Frequency Based Method:

The significance of term evaluation is given by the principle TFI X IDFI, where TFI is the frequency of term I in the document and IDFI is the inverted frequency of documents in which that term occurs. This method applied very simple strategy to generate abstract using TF/IDF based sentence extraction for single document summarization and use of single document summarization for multi-document. The procedure includes two phases such as: word hierarchical representation on the basis of most frequent terms in top of hierarchy and summarization based on hierarchical representation. Using simple statistical measures, Kernel is identified as the most significant passage of the source text that contains most frequent terms. It serves as the guideline to choose the other sentences for summary. (G. Salton, 1989; Jun'ichi Fukumoto, 2004; You Ouyang, 2009; Mr.Vikrant Gupta, 2012)

2.2.4 Text Summarization Techniques: A Brief Survey

In this paper emphasizes various extractive approaches for single and multi-document summarization. It describes some of the most extensively used methods such as topic representa-

tion approaches, frequency-driven methods, graph-based and machine learning techniques. Although it is not feasible to explain all diverse algorithms and approaches comprehensively in this paper, it provides a good insight into recent trends and progresses in automatic summarization methods and describes the state-of-the-art in this research area.

2.2.4.1 Extractive Summarisation:

As mentioned before, extractive summarization techniques produce summaries by choosing a subset of the sentences in the original text. These summaries contain the most important sentences of the input. Input can be a single document or multiple documents. In order to better understand how summarization systems work, here three fairly independent tasks which all summarizers perform

1) Construction intermediate representation of the input text which expresses the main aspects of the text.

2) Score the sentences based on the representation.

3) Select a summary comprising of a number of sentences.

There are two types of approaches based on the representation: topic representation and indicator representation.

Sentence Score When the intermediate representation is generated, that assigns an importance score to each sentence. In topic representation approaches, the score of a sentence represents how well the sentence explains some of the most important topics of the text. In most of the indicator representation methods, the score is computed by aggregating the evidence from different indicators. Machine learning techniques are often used to find indicator weights.

Summary Sentences Selection Eventually, the summarizer system selects the top k most important sentences to produce a summary. Some approaches use greedy algorithms to select the important sentences and some approaches may convert the selection of sentences into an optimization problem where a collection of sentences is chosen, considering the constraint that it should maximize overall importance and coherency and minimize the redundancy.

2.2.4.2Topic Representation Approaches:

2.2.4.2.1 Topic Words: This method is by using frequency thresholds to locate the descriptive words in the document and represent the topic of the document.

2.2.4.2.2 Frequency-driven Approaches: When assigning weights of words in topic representations, system can think of binary (0 or 1) or real-value (continuous) weights and decide which words are more correlated to the topic. The two most common techniques in this category are : word probability and TFIDF (Term Frequency Inverse Document Frequency).

2..2.4.2.3 Word Probability : The simplest method to use frequency of words as indicators of importance is word probability. The probability of a words is determined as the number of occurrences of the word

$$P(w) = \frac{\text{f(w)}}{\text{N}} \qquad \text{------ (2.8)}$$

where

f (w)= divided by the number of all words in the input

N is no. of documents

2.2.5 Extraction Based Multi Document Summarization using Single Document Summary Cluster:

Summarization is a reductive transformation of source text to summary text through content reduction by selection and/or generation on what is important in source text. Summarizing documents of all kinds of information is continually increasing and it is continued to be a steady subject of research over decades. This process of automatic summarization deals with preprocessing documents, evaluating the importance of sentences, generating summaries, evaluating summarization, and so on

The major challenge in multi-document summarization is that a document set may contain diverse information, which is either related or unrelated to the main central topic, and hence the system needs effective summarization methods to analyze and extract the important information. Additionally these information overlaps with each other, hence it needs effective merging techniques to build summary. In order to present the summary readable and inter–related with other

sentences, function of cohesion is studied. Cohesion relates part of a text to another part of the same text. Consequently it lends continuity to the text by providing this kind of text continuity. It also enables the reader or listener to ensure continuity in reading the document.

In order that effective summaries are to be built from multi document clusters, there exist two different approaches. The first approach extracts sentences from multi document clusters, while the next approach is to merge sentences extracted by single document approach. Consider an example to illustrate the need or importance of the proposed investigations. If a cluster C1 has 10 documents and each document having 10 sentences. If 10% compression ratio is applied, then the user needs to pick up 10 sentences (out of 100) from the cluster set. On analyzing the performance of such approach, it is found that summarizer tends to select sentences biased towards a document and tends to be repetitive. Hence this paper addresses this issue effectively to form multi document summary set from single document summaries. Also studies were made on the compression rates.

2.2.5.1.Centroid-Based Summarization (CBS)

The technique used for multi-document summarization is centroid-based summarization (CBS). CBS uses the centroids of the clusters produced by TDT to identify sentences central to the topic of the entire cluster. CBS is implemented in MEAD, which is publicly available multi-document summarizer.A key feature of MEAD is its use of cluster centroids, which consist of words that are central not only to one article in a cluster, but also to all the articles.

2.2.5.1.1 MEAD Extraction algorithm

MEAD compresses a cluster of topically related documents into a summary of the user's desired length. As a first step, three features namely centroid score,

position, and overlap with first sentence (which may happen to be the title of a document) is calculated:

• Centroid score- measure of the centrality of a sentence to the overall topic of a cluster (or document in the case of a single-document cluster).

• Position score- decreases linearly as the sentence gets farther from the beginning of a document.

• First sentence overlap score - which is the inner product of the TF*IDFweighted vector representations of a given sentence and the first sentence (or title, if there is one) of the document). As next step, the sentences are ranked according to their combined score which is a linear combination of all the sentence features used. MEAD uses a cosine similarity metric to compare each candidate sentence (for inclusion in the summary) to each higher-ranking sentence. If the candidate sentence is too similar to the specified threshold, it is penalized and is not included in the summary. Finally, the top remaining n-percent of the sentences (with the compression rate 'n' being determined by the user), are returned to the user as the summary.

2.2.5.1.2 Cluster-Based Relative Utility (CBRU) :

Cluster-based relative utility (CBRU, or relative utility, RU in short) refers to the degree of relevance (from 0 to 10) of a particular sentence to the general topic of the entire. A utility of 0 means that the sentence is not relevant to the cluster and a 10 marks an essential sentence.

2.2.5.1.3 Cross-Sentence Informational Subsumption (CSIS) :

A related notion to RU is cross-sentence informational subsumption (CSIS, or subsumption). CSIS reflects that certain sentences repeat some of the information present in other sentences and therefore omitted during summarization. If the information content of sentence a

$$i(a)<i(b) \qquad \text{------ (2.9)}$$

where

i(a) is contained within sentence b, then a becomes informationally redundant and the content of b is said to subsume that of a

2.2.5.2 Experimental Results and Analysis:

Comparison of MEAD and our approach based on summary generated from single document cluster.

Brief look on the sequence of steps carried out during the summary generation process of our system. The steps involved in the proposed approach are given below:

2.2.5.2.1Pre-processing the documents:

Preprocessing of documents involves several steps, each of which is explained in subsections that follow.

i. Removal of stop words:

Stop words are frequently occurring, insignificant words that appear in a database record, article or web page. Stop words are an application dependent. They apply to the particular database or application (e.g.: searching, summarization). It is commonly assumed that words, which are not members of the noun-verb adjective classes, should be on stop words lists. When a document is summarized by sentence extraction method System assigns weights to all the keywords or tokens in the input document. The process of doing such stop word elimination results in better summary generation. Since it eliminates these stop words unwanted sentences would never climb higher up the order. Single characters, common two-character and three-character words, frequently repeated words are typically included in the stop word list to maximize performance of summarization process.

ii. Applying Porter Stemming algorithm:

Truncation, also called stemming, is a technique that allows us to search for various word endings and spellings simultaneously. Stemming algorithms are used in many types of language processing and text analysis systems, and are also widely used in summarization, information retrieval and database search systems. A stemmer is a program determines a stem form of a given word. In other words, generates the morphological root of the word. Terms with a common stem will usually have similar meanings. For the example shown below the common root word is ‘IMPROV’.

IMPROVE, IMPROVED, IMPROVEMENTS

The suffix stripping process will reduce the total number of terms in the IR system, and hence reduce the size and complexity of the data in the system, which is advantageous.

2.2.5.2.2 Generation of Single document summary:

Consider an example to illustrate the summary generation process. If there are two documents in a cluster say document A & B, each having 10 sentences each correspondingly. For

10%, 20% and 30% compression rates System has to pick up 2, 4 and 6 sentences respectively. Picking up these sentences from multi document cluster is challenging (since it has 20 sentences). The sentence extraction algorithm generally applies statistical techniques to generate summary. In our extraction process, sentences are scored based on the term frequency. If the term matches the title words then special weight is given to those terms. The above approach is adopted for giving importance to title terms .Note that sentences are scored after removal and stop words and stemming the samples. System has not focused features like bold, Italics, Uppercase letters features for special weights. An important aspect that is to be discussed is whether a document should have a title or not. Once each sentence is scored those sentences are ranked based on the descending order of weights.

Each sentence is scored based on the frequency of 'n' terms occurring in the document (i.e TF). If the term matches the title of the document, then each term that matches the title is multiplied by title factor of 2. This special weight is not equal to first sentence overlap. The single document summarizer process single document at a time and generated summary. Term Frequency is calculated using expression. Sentence score has been obtained by adding obtaining the cumulative sum of Term Frequency.Based on the scores generated, sentences are chosen for summary depending on compression ratio.

2.2.5.2.3Merging Single document summaries:

Algorithm to merge summaries:

Step 1:Input: Set of single documents

Set of files :: Filei

compression ratio :: r

Step 2:Output : Merged list of sentences depending on multi document clusters;

Step 3:begin

Step 4:list ← empty; /* initially merged list is empty

Step 5:extract the sentences from each file depending on 'r'

Step 6:similarity ( ); /* measures the similarity of each sentence

Step 7:sort( ); / * sort the sentences based on score

Step 8:repeat( ) / * repeat for all files;

Step 9:merged_list <-merged_list + nn ; /* merge the sentences

Step 10:end;

2.3 EXTRACTIVE METHODS IN TEXT SUMMARSATION :

2.3.1 Term Frequency-Inverse Document Frequency (TF-IDF): The value increases proportionally to the number of times a word appears in the document.This method mainly works in the weighted term-frequency and inverse sentencefrequency paradigm ,where sentence-frequency isthe number of sentences in the document that contain that term. These sentence vectors are then scored by similarity to the query and the highest scoring sentences are picked to be part of the summary.Summarization is query-specific.

The hypothesis assumed by this approach is that if there are ‘‘more specific words’’ in a given sentence, then the sentence is relatively more important. The target words are usually noun.This method performs a comparison between the term frequency (tf) in a document -in this case each sentence is treated as a document and the document frequency (df), which means the number of times that the word occurs along all documents.

The TF/IDF score is calculated as follows:

$$\frac{TF}{IDF}(W) = DN(\frac{\log(1+tf)}{\log(df)}) \qquad \text{-----(2.10)}$$

where DN = number of documents

2.3.2 Cluster based method:

In this method, the semantic nature of a given document is captured and expressed in natural language by a set of triplets (subjects, verbs, objects related to each sentence).Cluster these triplets using similar information. The triplets statements are considered as the basic unit in the process of summarization. More similar the triplets are, the more the information is useless repeated; thus, a summary may be constructed using a sequence of sentences related the computed-clusters.

2.3.3 Graph theoretic approach:

In this technique, there is a node for every sentence. Two sentences are connected with an edge if the two sentences share some common words, in other words, their similarity is above some threshold. This representation gives two results: The partitions contained in the graph (that is those sub-graphs that are unconnected to the other sub graphs), form distinct topics covered in the documents. The second result by the graph-theoretic method is the identification of the important sentences in the document. The nodes with high cardinality (number of edges connected to that node), are the important sentences in the partition, and hence carry higher preference to be included in the summary. The graph theoretic method may also be adapted easily for visualization of inter and intra document similarity

2.3.4. Machine Learning approach:

In this method, the training dataset is used for reference and the summarization process is modeled as a classification problem: sentences are classified as summary sentences and non-summary sentences based on the features that they possess. The classification probabilities are learnt statistically from the training data, using Bayes' rule:

$$P(s \in <S| F1,F2,\ldots,FN) = P(F1,F2,,\ldots,FN|s \in S)*P(s \in S)/P(F1,F2,\ldots FN) \quad \text{-----(2,11)}$$

where,

s is a sentence from the document collection, F1, F2...FN are features used in classification.

S is the summary to be generated,

P (s∈< S | F1, F2, ..., FN) is the probability that sentence s will be chosen to form the summary given that it possesses features F1,F2...FN.

2.3.5. Text summarization with neural networks:

In this method, each document is converted into a list of sentences. Each sentence is represented as a vector [f1,f2,...,f7], composed of 7 features.

Seven Features of a Document:

1) f1 Paragraph follows title

2) f2 Paragraph location in document

3) f3 Sentence location in paragraph

4) f4 First sentence in paragraph

5) f5 Sentence length

6) f6 Number of thematic words in the sentence

7) f7 Number of title words in the sentence

The first phase of the process involves training the neural networks to learn the types of sentences that should be included in the summary. Once the network has learned the features that must exist in summary sentences, it needs to discover the trends and relationships among the features that are inherent in the majority of sentences. This is accomplished by the feature fusion phase, which consists of two steps:

1 Eliminating uncommon features

2. Collapsing the effects of common features.

2.3.5 Automatic Text Document Summarization:

2.3.5.1 Introduction:

A huge amount of data is available over internet. A large amount of data is uploaded over internet every day, which causes the availability of bulk data here. That means system has a large amount of data that will successfully match our search. Also it cannot forget the fact that a large amount of data is also present there which is not suitable for our search. In that case searching out a relevant data that meet our requirements is a tedious and time consuming task. There are two kinds of problems Searching a relevant document corresponding to our search.

Text summarization techniques can also be classified on the basis of volume of text documents available in the text database. Single-document summarization can only distill one document into a shorter version, whereas; multi-document summarization can condense a set of documents. Multi-document summarization can be seen as an enrichment of single-document summarization and can be used for outlining the information contained in a cluster of documents

2.2.5.2 Background System:

Anti- redundancy methods are needed since the degree of repetition as previously remarked is considerably higher in a group of topically related articles that in a sole article as each article tends to explain the main point as well as necessary shared background. The group of articles may contain a temporal dimension, typical in a stream of news reports about an unfolding event, in which case later

information may override earlier incomplete reports. The summary size required by the user will typically be much smaller for collections of topically related documents than for single documents requiring a lower compression factor (i.e. the size of the summary with respect to the size

of the document set), thereby requiring a far more careful selection of passages. The co-reference issue presents a greater challenge when entities and facts occur across documents than in a single-document situation.

Anti- redundancy methods are needed since the degree of repetition as previously remarked is considerably higher in a group of topically related articles that in a sole article as each article tends to explain the main point as well as necessary shared background. The group of articles may contain a temporal dimension, typical in a stream of news reports about an unfolding event, in which case later information may override earlier incomplete reports. The summary size required by the user will typically be much smaller for collections of topically related documents than for single documents requiring a lower compression factor (i.e. the size of the summary with respect to the size of the document set), thereby requiring a far more careful selection of passages. The co-reference issue presents a greater challenge when entities and facts occur across documents than in a single-document situation.

2.3.5.3 Proposed System:

A. Selection of Text Documents

In the first step text documents which are required to be summarized are given by the user.

B. Append and Tokenization

Text documents are appended and then the file content is tokenized into individual word.

C. Stemming and Removal of Stop Words

It finds out the root/stem of a word. Various suffixes are removed; number of words is reduced by having exactly matching stems. Language specific functional words which carry no information are removed.

D. Generation of List of Frequent Words

After eliminating stop words the term-frequent data and inverse document frequency is calculated from text documents and frequent terms are selected which are used to generate text document summary.

E. Sentence Generation

Similarity measure is evaluated using cosine algorithm and important sentences are generated. Unique sentences are clustered and re-sort. Finally, the summary is generated.

F. Update Details in Database

When the summary is generated then its details is stored in the database and is available to the user for information analysis.

G. Setup Web Service

A web service to provide summary of given text documents, will be set up. The Web Service client will send request message consisting of document then the server sends the summary as the response message.

2.2.5.4.Multiple Text Document Summarisation Algorithm:

Input:

1. Multiple text documents for which summary is to be generated.
2. Value of N for generation of N lines of summary.

Output:

1. Summary for text documents.
2. Compression Ratio.
3. Retention Ratio.

Steps:

1. Data Pre-processing Phase
   1.1. Retrieve text documents
   1.2.Apply stemming
   1.3.Eliminate stop words.
2. For the entire text content
   2.1.Get the TF and IDF
   2.2. Evaluate similarity measure
   2.3. Re-sort the important sentences
   2.4.Add sentences to summary-sentence-list
3. Calculate Compression Ratio and Retention Ratio

2.4 Extraction Methods

2.4.1Sentence length: This feature is the number of words present in the sentence. Longer sentences usually contain more information about the documents.

2.4.2 Named Entities count: Sentences which contain named entities are usually more important as these sentences indicate information of the entities participating in the documents.

This feature is a count of the named entities present in the sentence as indicated by the Stanford NER library4.

2.4.3 Top-K Important words: The TF-IDF sum feature might lead to the selection of long sentences with many insignificant words. To prevent this, the Top K Important words feature was introduced. It counts the number of words of this sentence present that are present in the top K words ranked by their TF-IDF scores.

2.4.4 Sentence Position: News articles tend to contain most of the important sentences in the first paragraph itself. Articles which are opinions by individuals tend to contain the summaries at the end of the document in a concluding paragraph. Hence sentence position tends to be a good indicator of the importance of a sentence across different classes of documents.

2.4.5 Numerical Literals count: Sentences which contain numerical literals usually indicate attributes of the events like death toll, time of occurrence, statistical information etc. This feature counts the number of Numerical terms present in the sentence.

2.4.6 Upper case letters count: Words which contain upper case letters are usually entities and hence system is using this feature to count the number of such instances.

2.4.7 Nouns count: Represents the number of noun classes in the sentence.

2.4.8 Verbs count: The count of the number of verbs and its various forms in the sentence.

2.4.9 Adjectives count: The count of the number of adjectives in the sentence.

2.4.10 Word Frequency :The thought of utilizing word frequency is that essential words seem commonly in the document. The most well-known measure broadly used to compute the saying recurrence is tf and idf.

2.4.11 Sentence location: Important data in a report is frequently secured by authors at the starting of the article. Therefore, the starting sentences are expected to contain the most imperative substance.

2.4.12Title / headline word: Occurrence of words from the report title in sentence demonstrates that the sentence is exceedingly important to the document.

2.4.13 Cue word: There are sure words in a sentence which demonstrate that the sentence is convey a useful message in the document (e.g., "essentially", "in conclusion").

2.4.14 Proper Noun: Sentences containing proper noun, place or thing speaking to a special element suchlike name of an individual, association or place are considered vital to the record.

CHAPTER 3

METHODOLOGY

3.1 Overall system architecture of text summarization

The proposed framework consists of 2 phases. The first phase is to apply preprocessing steps to all the documents. In the second phase feature extraction techniques like TF-IDF, is applied to summarize them into a single document.

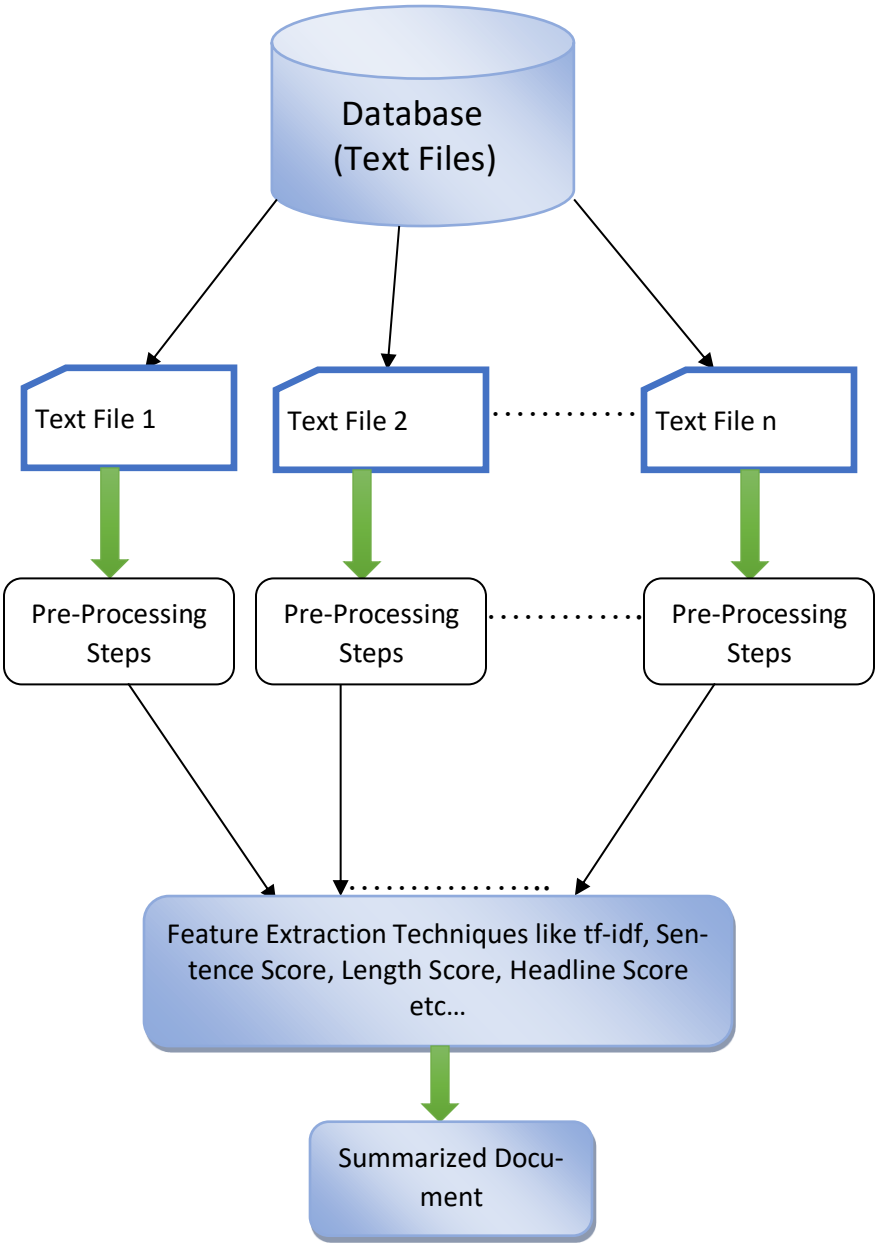

Figure 3.1 Architecture of the Proposed System

3.1.1 Text processing

In computing, text processing is the automated mechanization of the creation or modification of electronic text. Computer techniques are involved in text processing, which help in creating new content or bringing changes to content, searching or replacing content, formatting the content or generating a refined report of the content. It is concerned with automatic transmission of information. It deals with text processing utilities rather than text editing utilities.

3.1.2 Preprocessing

Data preprocessing describes any type of processing performed on raw data to prepare it for another processing procedure. Commonly used as preliminary data mining practice, data preprocessing transforms the data into a format that will be more easily and effectively processed for the purpose of the user. There are different steps and methods used for preprocessing of text, including: Stop Word Removal, Stemming.

3.1.2.1 Text Normalization:

Text normalization is a process of transforming text into a single canonical form. Normalizing text before storing or processing it allows for separation of required data from the rest so that the system can send consistent data as an input to the other steps of the algorithm.

3.1.2.2 Stop Word Removal

Stop Word: A Stop Word is a commonly used word in any natural language such as “a, an, the, for, is, was, which, are, were, from, do, with, and, so, very, that, this, no, yourselves etc….”.(Appendix A)

These Stop Words will have a very high frequency and so these should be eliminated while calculating the term frequency so that the other important things are given priority. Stop word removal is such a Pre-processing step which removes these stop words and thereby helping in the further steps and also reducing some processing time because the size of the document decreases tremendously.

3.1.2.3 Stemming:

Stemming is a pre-processing step in Text Mining applications as well as a very common requirement of Natural Language processing functions. In fact it is very important in most of the Information Retrieval systems. The main purpose of stemming is to reduce different grammatical forms / word forms of a word like its noun, adjective, verb, adverb etc. to its root form. The goal of stemming is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form.

Eg: A stemmer for English should identify the strings "cats", "catlike", "catty" as based on the root "cat".

In multiple document summarization the following preprocessing steps have been done.

3.1.3Proposed System For Text Summarization

3.1.3.1 Feature Extraction Algorithm

STEP1:

compute

For each document in the articles:

do:

Divide the document into sentences

end

STEP2:

compute

For each document in the articles:

do:

remove stop words

end

STEP 3:

compute

For each sentence in the document:

do:

divide the sentence into tokens and apply stemmer

end

STEP 4:

compute

For each sentence in the document:

do:

For each word in the sentence:

do:

calculate word probability

$$wp = \frac{\text{wf}}{\text{tw}} \qquad \text{-------(3.1)}$$

Where

wp=word probability

wf = word frequency

tw = total words in the document

end

end

STEP 5:

compute

For each sentence in the document:

do:

calculate sentence frequency

$$\text{sf} = \frac{\sum_{i=0}^{n} \text{wp}}{ls} \qquad \text{--------(3.2)}$$

Where

Sf = sentence frequency

n = total number of words in the sentence

wp = word probability

ls = length of the sentence

end

STEP 6:

Compute

For each sentence in the document:

Do:

Calculate Length Score using the formula

$$ls = \frac{(\text{sl}-\text{il})}{il} \qquad \text{-------(3.3)}$$

Calculate Headline Score and Position Score hs & ps

end

STEP 7:

Compute

For each sentence in the document:

Do:

Calculate sentence score using the formula:

$$ss = \frac{(\text{tf}-\text{idf}+\text{ls}+\text{ps}+\text{hs})}{4} \qquad \text{-------(3.4)}$$

Where

ss = Sentence Score

tf-idf = tf*idf

tf = Term Frequency

idf = InverseDocument frequency

ls= Length Score

ps = Position Score

hs = HeadLine Score

end

STEP 8:

Get the compression ratio and using that compute the number of lines to be present in the output.

N= CR * TL -------(3.5)

where

N= No. of lines in the output

CR= Compression ratio

TL= Total number of lines in all documents.

STEP 9:

Get the top n sentences from both the documents and print them in output.

STEP 10: Stop

3.1.3.2 Algorithm Illustration:

Step 1: Document 1 is divided into sentences.

Table 3.1: Sample input Document1

Sentence no	Text
1	Much of the Chinese capital shut down Tuesday after Beijing's city government issued its first red alert for pollution, closing schools and construction sites and restricting the number of cars on the road.
2	Beijing's Municipal Bureau of Environmental Protection warned that severe pollution would affect the Chinese capital for several days, starting Tuesday morning.
3	According to the U.S. Embassy in Beijing, the air quality index stood at 250 Tuesday morning, classed as "very unhealthy" and 10 times higher than the World Health Organization's recommended levels.
4	Gao Yuanli, 35, told CNN that the smog often made her life difficult.
5	She wears face masks on most winter days, and she bought an air purifier two years ago, she said.
6	"I can't go out on weekends now if the air is bad, and I don't go to outdoor markets anymore," she said.
7	The alert means extra measures will be enforced.

8	Car use is being cut in half by having only odd- or even-numbered license plates on the road at any one time.
9	Heavy vehicles, including garbage trucks, are banned from the streets.
10	Other polluting industrial activity has been curbed, as have fireworks and outdoor barbecuing.
11	The red alert -- the highest level in the system -- is due to be in force until noon Thursday local time.
12	The city's roads and sidewalks were much quieter than usual Tuesday, and small-business owners like Jia Xiaojiang, who makes egg pancakes, complained of fewer customers.
13	Jia doesn't wear face masks but says the pollution has caused her respiratory distress.
14	"The smog is like toxic gas," she said.
15	"I never had a sore throat before."
16	Starting from last year, my throat hurts once I speak."
17	The red alert caused disruption for some parents, who had to scramble Monday evening to find alternative childcare arrangements.
18	Li Ning, a 33-year-old IT worker, said his child was being looked after by grandparents.
19	CNN reporters in Beijing said the pollution didn't feel as severe as last week, when air quality, as measured by the U.S. Embassy, went above 500 or "beyond index" Monday and Tuesday.
20	Some residents have questioned why the unprecedented red alert level was not issued then.
21	But others are resigned to living with pollution that is regularly 10 times worse than recommended levels

22	"I'd find a day when the sky is blue unusual," said Wolf Hu.
23	He travels often for work, often preferring China's high-speed rail network to flying, which is prone to smog-related delays -- 12 outbound flights and 14 in-bound flights into the city were canceled Tuesday.
24	According to the state-run news agency Xinhua, a red smog alert is issued only when heavy pollution is expected to last longer than 72 hours.
25	China is the world's largest emitter of greenhouse gases.
26	It aims to cut is peak emissions in half by 2030.
27	Most of the country's carbon emissions come from burning coal to heat homes and fuel power plants, a practice that spikes during winter months.

Table 3.2: Sample input Document2

Sentence no	Text
1	Red has been considered the color of prosperity and good fortune in China for centuries, and it is also the color of the Communist Party.
2	But this week, it took on a darker meaning here, as it began to symbolize the failure of the party to rein in toxic smog that regularly endangers the health of hundreds of millions of people in the country’s north.
3	The Beijing government sounded its first-ever air pollution red alert on Monday night, prompting many of the city’s 22 million residents to take precautions through Thursday, when strong winds blew the smog away.
4	The emergency measures ended at noon.
5	The alert was another of the touchstone moments that have occurred regularly since 2012, when the party began relaxing its tight control over information on air quality.

6	Since then, crucial decisions made every few months by senior Chinese officials have broadened the public's understanding of the environmental degradation afflicting the nation, and they have given people more tools to gauge methods for protecting themselves.
7	But those decisions have also raised questions about whether the party is up to the herculean challenge of cleaning up China's environment.
8	On no other issue are President Xi Jinping and other Chinese leaders forced to walk such a fine line, between controlling information that has the potential to undermine their legitimacy and doling it out to increasingly anxious citizens who consider such disclosures essential.
9	And as awareness of their toxic environment grows, people are demanding fundamental solutions, not just periods of high alert that lead to inconveniences like school closings.
10	The emergency measures did not even achieve what they were intended to do: Despite factory shutdowns and strict traffic controls, the smog remained severe in Beijing until the strong winds blew it away.
11	That in turn led to millions south of the capital suffering from even more polluted air.
12	"I don't care for the alert system," said Kan Tingting, a cafe manager who had stayed at home with her 3-year-old daughter on Tuesday.
13	"It's rather pointless, if you ask me, because it doesn't solve any real problems.
14	No real progress is made until all the factories move away and the odd-even license plate number restriction becomes permanent."
15	Ms. Kan was referring to the driving limitations imposed as part of the emergency measures, which were supposed to keep about half of Beijing's five million cars off the streets.
16	While some motorists complained about, and even violated, the rule, others like Ms. Kan said they wanted it to be made permanent, to reduce pollution levels and free up Beijing's clogged roads.

17	Scholars and environmental campaigners echoed Ms. Kan's sentiments, saying that while the party's progress in environmental transparency had benefited ordinary Chinese, the real test was whether it could tackle the roots of the problem, which, in the case of air pollution, is industrial coal use, a crucial component of the nation's rapid economic growth.
18	Vehicle emissions are another big source of pollutants.
19	"The alert system is just a stopgap measure — giving the public the ability to protect themselves from pollution and taking emergency measures to quickly reduce pollution," said Alex Wang, a law professor at the University of California, Los Angeles, who studies China's environmental policy.
20	"More extensive changes in the regulatory system — some already underway — are needed to fix the problem for good."In the long term, accurate monitoring data is the foundation of regulatory changes needed to reduce pollution and protect human health."
21	Mr. Wang said the positive moves by officials regarding transparency and the reporting of pollution data were necessary to the evolution of the regulatory system.
22	"China greatly expanded its disclosure of air quality data a few years ago in response to public outrage at extreme pollution levels in many cities," he said.
23	"This raised public awareness and made 'PM 2.5' a household word in China.
24	In the long term, accurate monitoring data is the foundation of regulatory changes needed to reduce pollution and protect human health."
25	PM 2.5 refers to fine, deadly particulate matter that can enter the bloodstream through the lungs.
26	Beijing began releasing real-time data on PM 2.5 to the public in 2012, after years of pressure from prominent residents who used online platforms to get their message across.

27	One crucial figureure in that push was Pan Shiyi, a real estate mogul, who repeatedly asked in public why the Chinese government was keeping its citizens in the dark on PM 2.5 while the United States Embassy in Beijing shared such data hourly on a Twitter feed.
28	In January 2013, when a thick haze smothered northern China and the word "airpocalypse" was coined, the outcry was so loud that officials began allowing state news media to report more widely on air pollution.
29	Another official calibration came in February, when Chai Jing, a former investigative reporter for the state-run China Central Television, posted online a searing documentary about toxic air that she had made with some former journalist colleagues.
30	The video, which was made with the cooperation of some officials, got hundreds of millions of views within days, and it was praised by the new environmental minister — before censors had it taken offline.
31	Beijing's air quality in the first half of this year improved from the same period in 2014, with average PM 2.5 levels dropping 15 percent, according to the state news agency Xinhua.
32	This week's red alert was the latest step seeking to loosen up discussion of environmental hazards and to allow citizens to vent their frustration.
33	Beijing officials have even thanked the city's residents for their response, writing in an open letter on Thursday, "The dedication and full support of the people of Beijing touched us deeply."
34	Officials raised the alert just one week after being widely criticized for inaction during a multiday spell of foul air that descended on northern China as Mr. Xi met other leaders in Paris for climate change talks.
35	People question whether the Beijing government will continue to raise the red alert in accordance with an air crisis policy announced in 2013 and revised this year.
36	The policy requires officials to do so whenever the air quality index is forecast to rise above 200 for 72 straight hours.

37	If officials had declared code red every time since 2013 the index met that criterion, they would have done so eight times, for a total of 36 days, according to data analysis by the local makers of a popular air quality phone app.
38	Officials do have an out — they can say the predictions of smog are too fuzzy for them to raise the alarm.
39	“In the future, they might say we’re not that confident in the forecast,” said Wang Tao, an energy and climate scholar at the Carnegie – Tsinghua Center for Global Policy in Beijing.
40	But the mayor of Beijing, Wang Anshun, has acknowledged that public environmental awareness, while perhaps leading to more criticism of China’s development path and weak regulatory efforts, is needed to help solve the crisis.
41	“We must take effective measures and enforce them with no reductions,” he said at a meeting on Dec. 4, according to an official news report.
42	“We must accept supervision from the public and the media, in order to win the battle against the imminent heavy air pollution.”

Step 2: Stop Words are removed after each and every sentence and the remaining words of document 1 are shown in table 3.3

Table 3.3: List of words for Document 1 after Stop Word Removal

Sentence number	List of words
1	much, of, the, chines, capit, shut, down, tuesday, after, beij, citi, govern, issu, it, first, red, alert, for, pollut, close, school, and, construct,site,and,restrict,the,number,of,car, on, the, road
2	beij, s, municip, bureau, of, environment, protect, warn, that, sever, pollut, would, affect, the, chines, capit, for, sever, day, start, tuesday, morn
3	accord, to, the, u.s., embassi, in, beij, the, air, qualiti, index, stood, at, tuesday, morn, class, as, veri, uunhealthi, and, time, higher, than, the, world, health, organ,

	s, recommend, level
4	gao, yuan, told, cnn, that, the, smog, often, made, her, life, difficult
5	she, wear, face, mask, on, most, winter, day, and, she, bought, an, air, purifi, two, year, ago, she, said
6	i, ca, n't, go, out, on, weekend, now, if, the, air, is, bad, and, i, do, n't, go, to, out-door, market, anymor, she, said
7	the, alert, mean, extra, measur, will, be, enforc
8	car, use,is, be, cut, in, half, by, have, onli, odd-, or, even-numb, licens, plate, on, the, road, at, ani, one, time
9	heavi, vehicl, includ, garbag, truck, are, ban, from, the, street
10	other, pollut, uindustri, activ, has, been, curb, as, have, firework, and, outdoor, barbecue
11	the, red, alert, the, highest, level, in, the, system, is,due, to, be, in, forc, until, noon,thursday, local, time
12	the, citi, s, road, and, sidewalk, were, much, quieter, than, usual, tuesday, and, small-busi, owner, like, jia, xiaojiang, who, make, egg, pancak, complain, of, fewer, custom
13	jia, doe,n't, wear, face, mask, but, say, the, pollut, has, caus, her, respiratori, dis-tress
14	the, smog, is, like, toxic, gas, she, said
15	i, never, had, a, sore, throat, befor
16	start, from, last,year, my, throat, hurt, one,i, speak
17	the, red, alert, caus, disrupt, for, some, parent, who, had, to, scrambl, monday, even, to, find, altern, childcar, arrang

18	li, ning, a, 33-year-old, it, worker, said, his, child, was, be, look, after, by, grand-par
19	cnn, report, in, beij, said, the, pollut, did, n't, feel, as, sever, as, last, week, when, air, qualiti, as, measur, by, the, u.s, embassi, went, abov, or, beyond, index, mon-day, and, Tuesday
20	some, resid, have, question, whi, the, unpreced, red, alert, level, was, not, issu, then
21	but, other, are, resign, to, live, with, pollut, that, is, regular, time, wors, than, rec-ommend, level
22	i, d, find, a, day, when, the, sky, is, blue, unusu, said, wolf, hu
23	he, travel, often, for, work, often,prefer, china, s, high-spe, rail, network, to, fli, which, is, prone, to, smog-rel, delay, outbound, flight, and, inbound, flight, into, the, citi,were, cancel, Tuesday
24	accord, to, the, state-run, news, agenc, xinhua, a, red, smog, alert, is, issu, onli, when, heavi, pollut, is, expect, to, last, longer, than, hour
25	china, is, the, world,s, largest, emitt, of, greenhous, gase
26	it, aim, to, cut, is,peak, emiss, in, half, by
27	most, of, the, countri, s, carbon, emiss, come, from, burn, coal, to, heat, home, and, fuel, power, plant, a, practic, that, spike, dure, winter, month

Table 3.4: List of words for Document 2 after Stop Word Removal

Sentence number	List of words
1	beij, red, has, been, consid, the, color, of, prosper, and, good, fortun, in, china, for, centuri, and, it, is, also, the, color, of, the, communist, parti
2	but, this, week, it, took, on, a, darker, mean, here, as, it, began, to, symbol, the, failur, of, the, parti,to, rein, in, toxic, smog, that, regular, endang, the, health, of, hundr, of, million, of, peopl, in, the, countri, north

3	the, beij, govern, sound, it, first-ev, air, pollut, red, alert, on, monday, night, prompt, mani, of, the, citi, million, resid,to, take, precaut, through, thursday, when, strong, wind, blew, the, smog, away
4	the, emerg, measur, end, at, noon
5	the, alert, was, anoth,of, the, touchston,moment, that, have, occur, regular, sinc, when, the, parti, began, relax, it, tight, control, over, inform, on, air, quality
6	sinc, the,n ucrucial, decis, made, everi, few, month, by, senior, chines, offici, have, broaden, the, public, understand, of,the, environment, degrad, afflict, the, nation, and, they, have, given,peopl, more, tool, to, gaug, method, for, protect, themselv
7	but, those, decis,have, also, rais, question, about, whether,the, parti, is, up, to, the',herculean, challeng,of, clean, up, china, environ
8	on, no, other, issu, are, presid, xi, jinp, and, other, chines, leader, forc, to, walk, such, a, fine, line, between, control, inform,that, has,the, potenti, to, undermin, their, legitimaci, and, do, it, out, to, increas, anxious, citizen, who, consid, such, disclosur, essenti
9	and, as, awar, of, their, toxic, environ, grow, peopl, are, demand, fundament, so-lut, not, just, period, of, high, alert, that, lead, to, inconveni, like, school, close
10	the, emerg, measur, did, not, even, achiev, what, they, were, intend, to, do, despit, factori, shutdown,and, strict, traffic, control,the, smog, remain, sever, in, beij, until, the, strong, wind, blew, it,away
11	that, in, turn, led, to, million, south,of, the, capit\, suffer, from, even, more, pol-lut, air
12	i, dont, care, for, the, alert, system, said, kan, tingt, a, cafe, manag, who, had, stay, at, home, with, her, 3-year-old, daughter, on, Tuesday
13	it, rather, pointless, if, you, ask, me, becaus, it, doesnt, solv, ani, real, problem
14	no, real, progress, is, made, until, all, the, factori, move, away, and, the, odd-even, licens, plate, number, restrict, becom, perman

15	ms., kan, was, refer, to, the, drive, limit, impos, as, part, of, the, emerg,measur, which, were, suppos, to, keep, about, half, of, beij, five, million, car, off, the, street
16	while, some, motorist, complain, about, and,even, violat, the, rule, other, like, ms., kan, said, they, want, it,to, be, made, perman, to, reduc, pollut, level, and, free, up, beij, clog, road
17	scholar, and, environment, campaign, echo, ms., kan, sentiment, say, that, while, the,part, progress, in, environment,transpar, had, benefit, ordinari, chines, the, real, test, was, whether, it, could, tackl, the, root, of, the, problem, which, in, the, case, of, air, pollut, is, industri, coal, use, a, crucial, compon, of, the, nation, rapid, econom, growth
18	vehicl, emiss, are, anoth, big, sourc, of, pollut
19	the, alert, system, is, just, a, stopgap, measur, give, the, public, the, abil, to, protect, themselv, from, pollut, and, take, emerg, measur, to, quick, reduc, pollut, said, alex, wang, a, law,professor, at, the, univers, of, california, los, angel, who, studi, china, environment, polici
20	more, extens, chang, in, the, regulatori, system, some, alreadi, underway, are, need, to, fix, the, problem,for, good
21	mr., wang, said, the, posit, move,by, offici, regard, transpar, and, the, report, of, pollut, data, were, necessari, to, the, evolut, of, the, regulatori, system
22	china, great, expand, it, disclosur, of, air,qualiti, data,a, few, year, ago, in, respons, to, public, outrag, at, extrem, pollut, level, in, mani, citi, he, said
23	this, rais, public, awar, and,made, pm, a, household, word, in, china
24	in, the, long, term, accur,monitor, data, is, the, foundat, of, regulatori, change, need, to, reduc, pollut, and, protect,human, health
25	pm, refer, to, fine, dead, particul, matter, that, can, enter, the, bloodstream, through, the, lung
26	beij, began, releas, real-tim, data, on, pm, to, the, public, in, after, year, of, pressur, from, promin, resid, who, use, onlin, platform, to,get, their, messag, across

27	one, crucial, figureur, in, that, push, was, pan, shiyi, a, real, estat, mogul, who, repeat, ask, in, public, whi, the, chines, govern, was, keep, it, citizen, in, the, dark, on, pm, while, the, unit, state, embassi, in, beij, share, such, data, hour, on,a, twitter, feed
28	in, januari, when, a, thick, haze, smother, northern, china,and, the, word, airpocalyps, was, coin, the, outcri,was, so, loud, that, offici, began, allow, state,news, media, to, report, more, wide, on, air, pollut
29	anoth, offici, calibr, came, in, februari, when, chai, jing, a, former, investig, report, for, the, state-run, china, central, televis, post, onlin, a, sear, documentari, about, toxic, air, that, she, had, made, with, some, former, journalist, colleague
30	the, video, which, was, made, with, the, cooper, of, some, offici, got, hundr, of, million, of, view, within, day, and,it, was, prais, by, the, new, environment, minist, befor, censor, had, it, taken, offlin
31	beij, air, qualiti, in, the, first, half, of, this, year, improv, from, the, same, period, in, with, averag, pm, level, drop, percent, accord, to, the, state, news, agenc, Xinhua
32	this, week, red, alert, was, the,latest, step,seek, to, loosen, up, discuss, of, environment, hazard, and, to, allow, citizen, to, vent, their, frustra
33	beij, offici, have, even, thank, the, citi,resid, for, their, respons, write, in, an, open, letter, on, thursday, the, dedic, and, full, support,of, the, peopl, of, beij, touch, us, deeply
34	offici, rais, the, alert, just, one, week, after, be, wide, critic, for, inact, dure, a, multiday, spell, of, foul, air, that, descend, on, northern, china, as, mr., xi, met, other, leader, in, pari, for, climat, chang, talk
35	peopl, question, whether, the, beij, govern, will, continu, to, rais, the, red, alert, in, accord, with, an,air, crisi, polici, announc, in, and, revis, this, year
36	the, polici, requir, offici, to, do, so, whenev, the, air,quality, index, is, forecast, to, rise, abov, for,straight, hour
37	if, offici, had, declar, code, red,everi, time, sinc, the, index, met, that, criterion, they, would, have, done, uso, eight, time, for, a, total, of, day, accord, to, data, analysi, by, the, local, maker, of, a, popular, air, qualiti, phone, app

38	offici,do, have,an, out, they, can, say, the, predict, of, smog, are, too, fuzzi, for, them, to, rais, the, alarm
39	in, the, futur, they, might, say, were, not, that, confid, in, the, forecast, said, wang, tao, an, energi, and, climat, scholar, at, the, carnegi, tsinghua, center,for, global, polici, in, beij
40	but, the, mayor, of, beij, wang, anshun, has, acknowledg, that, public, environment, awar, while, perhap, lead, to, more, critic, of, china, develop, path, and, weak, regulatori, effort, is, need, to, help, solv, the, crisi
41	we, must, take, effect, measur, and, enforc, them, with,no, reduct, he, said, at, a, meet, on, dec.,accord, to, an, offici,news, report
42	we, must, accept, supervis, from, the, public, and, the, media, in, order, to, win, the, battl, against, the, immin, heavi, air, pollut

Step 3: Probability of each word is calculated and is shown in table 3.5

Table 3.5: Calculation of probability for the list of words for Document 1

WORD	PROBABILITY	WORD	PROBABILITY	WORD	PROBABILITY
Chines	0.086782047003	Capit	0.0544462934482	shut	0.0446927024471
Tuesday	0.227019974611	Beij	0.262827728283	Road	0.142893156187
Citi	0.1091011888	govern	0.023328257473	issu	0.0775290925686
Red	0.22115386082	alert	0.324182890143	Pollut	0.393481823781
Close	0.0257924070488	school	0.0534345085784	con-struct	0.0310966820066
Site	0.0453614644479	restrict	0.0356501351149	number	0.0310966820066

Car	0.0807106909988	Road	0.142893156187	beij	0.262827728283
Capit	0.0544462934482	municip	0.0540304816906	bureau	0.0409744193753
Environ-ment	0.0476310520625	protect	0.0351182751382	warn	0.0371679004344
sever	0.0863386980626	pollut	0.393481823781	affect	0.0317867991329
Chines	0.086782047003	Capit	0.0544462934482	sever	0.0863386980626
day	0.0788511726874	Start	0.0564794604986	tuesday	0.227019974611
Morn	0.0691764504976	accord	0.0541025794332	u.s.	0.0362227282985
embassi	0.0962772899621	Beij	0.262827728283	air	0.155375542797
Quality	0.0786716934568	Index	0.0729243892018	stood	0.0389120391966
Tuesday	0.227019974611	Morn	0.0691764504976	class	0.038384904553
Very	0.0305299485119	Time	0.10055470155	higher	0.026594104773
World	0.0523951349922	health	0.0387095800243	organ	0.0348103864524
mask	0.121918621182	Recomm	0.0709640879611	level	0.101754235615

gao	0.0778240783932	Yuan	0.0528862664606	told	0.0220187313471
Cnn	0.125859503075	Life	0.0377205686396	difficult	0.0367557839706
Wear	0.116364260116	face	0.067660471664	mask	0.121918621182
winter	0.0778240783932	Day	0.0788511726874	bought	0.0295029529705
Air	0.155375542797	purifi	0.0657069320707	year	0.028485515252
Ago	0.0298954721799	Said	0.0601571726914	ca	0.0467164461525
car	0.0807106909988	weekend	0.0395580079319	air	0.155375542797
Bad	0.0398658966178	Cut	0.0514044330577	outdoor	0.125859503075
Market	0.0189531136711	anymor	0.0629297515376	said	0.0601571726914
Alert	0.324182890143	Mean	0.0334805400786	extra	0.0446927024471
measur	0.0608599118672	enforc	0.0463012056887	car	0.0807106909988
Use	0.0264039876054	Cut	0.0514044330577	half	0.0591804145421
Onli	0.0523097843115	Licens	0.043260555367	plate	0.0546832963734

road	0.142893156187	ani	0.0255837806679	time	0.10055470155
Heavi	0.0702365502765	vehicl	0.0450189245753	includ	0.0198831335268
truck	0.0463012056887	Ban	0.0436596606762	street	0.0394831506845
Pollut	0.393481823781	industri	0.0209148884455	activ	0.0311626475563
Curb	0.039191644138	odour	0.125859503075	red	0.22115386082

Table 3.6: Calculation of probability for the few words for Document 2

WORD	PROBABILITY	WORD	PROBABILITY	WORD	PROBABILITY
Red	0.111426399053	Consid	0.0296427804991	color	0.0586290939488
Prosper	0.0293145469744	Good	0.0139292714101	fortun	0.0251596441367
China	0.162046263585	Century	0.0243895770959	color	0.0586290939488
Communist	0.0263905272026	Parti	0.0895969850403	week	0.0338045396669
Took	0.0187535710426	Mean	0.0168688713164	began	0.0687163937191
Symbol	0.0263905272026	Failur	0.0197824965202	parti	0.0895969850403
Rein	0.0266462733646	Toxic	0.0921414138019	regular	0.0353485059727
End	0.0257204959756	Health	0.0195034764257	hundr	0.0448773651465
Million	0.10713170531	People	0.0879813055181	countri	0.0121279895371

Step 4: Sentence score is calculated and is shown in table 3.7, This is calculated by summing up all the probability scores of all the words in each sentence and dividing it with the length of the sentence

Table 3.7: Calculation of Frequency score for Document 1

Sentence No	Sentence Score	Sentence Length	Frequency Score
1	2.37311197233	34	0.069797410951
2	1.74561614874	22	0.0793461885791
3	1.68337237414	30	0.0561124124714
4	0.333422983551	12	0.0277852486293
5	0.831742192005	19	0.0437759048424
6	0.806898797713	24	0.033620783238
7	0.509517250225	8	0.0636896562781
8	0.636984800661	22	0.0289538545755
9	0.264582625428	10	0.0264582625428
10	0.610610506998	13	0.0469700389998
11	0.955059325005	20	0.0477529662503
12	0.984634005255	26	0.0378705386636
13	1.0687139964	15	0.0712475997603
14	0.205547343629	8	0.0256934179537
15	0.0854802706331	7	0.0122114672333
16	0.203817742431	10	0.0203817742431

17	0.886660262386	19	0.0466663295993
18	0.313044099211	15	0.0208696066141
19	2.03367763836	32	0.0635524261986
20	0.90149344231	14	0.0643923887365
21	0.834952380512	16	0.052184523782
22	0.475543462268	14	0.033967390162
23	1.22507538755	31	0.0395185608888
24	1.40169596689	24	0.0584039986203
25	0.375823091436	10	0.0375823091436
26	0.307709274718	10	0.0307709274718
27	0.824306673379	25	0.0329722669352

Table 3.8: Calculation of Frequency score for Document 2

SNO	SENTENCE SCORE	SENTENCE LENGTH	FREQUENCY SCORE
1	1.05953024995	26	0.0407511634598
2	0.742292159775	40	0.0185573039944
3	2.04379760425	32	0.0638686751328
4	0.207033447699	6	0.0345055746165
5	1.01144422538	26	0.0389017009763
6	1.1114232234	37	0.0300384654974

7	0.476526760044	22	0.0216603072747
8	0.616637991488	43	0.0143404184067
9	0.664346789901	26	0.0295465657029
10	1.05637081245	33	0.0320112367409
11	0.786264445851	16	0.0491415278657
12	0.50692296117	24	0.0211217900487
13	0.210193906378	14	0.0150138504556
14	0.331701521885	20	0.0165850760942
15	1.02708325617	30	0.0342361085391
16	1.21526233522	32	0.0379769479757
17	1.89776072607	54	0.0351437171494
18	0.470895238373	8	0.0588619047966
19	2.20275317122	44	0.0500625720733
20	0.293506679036	18	0.0163059266131
21	1.01228423225	25	0.0404913692899
22	1.44297774389	27	0.0534436201441
23	0.482179482894	12	0.0401816235745
24	0.833034475053	21	0.0396683083359
25	0.146412284622	15	0.00976081897483
26	0.948294086958	27	0.0351220032207

27	1.2966481121	46	0.028188002437
28	1.34329437342	34	0.0395086580417
29	0.920534728366	36	0.0255704091213
30	0.629600452954	34	0.018517660381
31	1.13052037483	29	0.0389834612011
32	0.781603498659	24	0.0325668124441
33	1.42313640464	31	0.0459076259562
34	1.33521427567	37	0.0360868723153
35	1.45703197769	26	0.0560396914495
36	0.735830066645	20	0.0367915033323
37	1.17201548305	41	1.17201548305
38	0.267199595387	21	0.01272379025
39	0.825961008012	31	0.0266439034843
40	1.48887201646	34	0.0437903534253
41	0.487670304688	24	0.0203195960287
42	0.906982736503	22	0.0412264880229

Step 5: Calculating the Headline Score for Document. Because the words in the Headlines are more important than the remaining. So, this gives more score to the words which are relevant to the headline.

Table 3.9 Calculation of sentence score for Document 1 based on header method.

Sentence no	Headline Score	Sentence No	Headline score
1	2.6	2	1.97777777778
3	2.31111111111	4	0.911111111111
5	1.13333333333	6	1.11111111111
7	0.666666666667	8	1.46666666667
9	1.0	10	1.17777777778
11	1.4	12	1.95555555556
13	0.933333333333	14	0.377777777778
15	0.444444444444	16	0.622222222222
17	1.57777777778	18	1.02222222222
19	2.15555555556	20	1.15555555556
21	1.31111111111	22	0.733333333333
23	2.62222222222	24	1.75555555556
25	0.733333333333	26	0.444444444444
27	1.71111111111		

Table 3.10: Calculation of sentence score for Document 2 based on header method

Sentence no	Headline Score	Sentence no	Headline Score
1	1.85106382979	2	2.70212765957
3	2.44680851064	4	0.510638297872
5	2.04255319149	6	3.17021276596
7	1.76595744681	8	3.3829787234
9	2.06382978723	10	2.36170212766
11	1.14893617021	12	1.51063829787
13	0.893617021277	14	1.55319148936
15	2.14893617021	16	2.23404255319
17	4.29787234043	18	0.659574468085
19	3.40425531915	20	1.3829787234
21	1.89361702128	22	1.78723404255
23	0.808510638298	24	1.74468085106
25	1.23404255319	26	2.08510638298
27	3.25531914894	28	2.36170212766
29	2.91489361702	30	2.23404255319
31	2.12765957447	32	1.65957446809
33	2.46808510638	34	1.00878653689
35	2.02127659574	36	1.36170212766

37	2.59574468085	38	1.21276595745
39	2.29787234043	40	2.70212765957
41	1.70212765957	42	1.44680851064

Step 6: Calculating the Length Score, Length Score is based on the ideal length of the sentence because too short sentences does not make much meaning, neither do the very long sentences.

Table 3.11: Calculation of score of each sentence in Document 1 using length method

Sentence no	Length score	Sentence No	Length score
1	0.65	2	0.05
3	0.55	4	0.35
5	0.05	6	0.1
7	0.6	8	0.1
9	0.5	10	0.35
11	0.0	12	0.25
13	0.3	14	0.6
15	0.65	16	0.5
17	0.05	18	0.25
19	0.6	20	0.3
21	0.15	22	0.35
23	0.6	24	0.25
25	0.55	26	0.45

27	0.2		

Table 3.12: Calculation of score of each sentence in Document 2 using length method

Sentence NO	Length score	Sentence No	Length Score
1	0.3	2	1.0
3	0.65	4	0.7
5	0.35	6	0.85
7	0.1	8	1.15
9	0.3	10	0.65
11	0.2	12	0.2
13	0.3	14	0.0
15	0.5	16	0.6
17	1.7	18	0.6
19	1.2	20	0.1
21	0.25	22	0.35
23	0.35	24	0.05
25	0.2	26	0.45
27	1.35	28	0.75
29	0.8	30	0.7
31	0.6	32	0.2
33	0.55	34	0.85

35	0.35	36	0.1
37	1.15	38	0.05
39	0.55	40	0.7
41	0.25	42	0.1

Step 7: Calculating Position score using the relative position of the sentence, Because the sentences which are either in the starting of the article or in the ending (Conclusion) are generally more important than the remaining. So, System gives higher weightage to those sentences

Table 3.13: Calculation of score of each sentence in Document 1 using position method

Sentence no	Position score	Sentence no	Position score
1	0.17	2	0.17
3	0.23	4	0.23
5	0.23	6	0.14
7	0.14	8	0.14
9	0.08	10	0.08
11	0.05	12	0.05
13	0.05	14	0.04
15	0.04	16	0.04
17	0.06	18	0.06
19	0.04	20	0.04
21	0.04	22	0.04
23	0.04	24	0.04

25	0.15	26	0.15
27	0.15		

Table 3.14: Calculation of score of each sentence in Document 2 using position method

Sentence no	Position score	Sentence no	Position score
1	0.17	2	0.17
3	0.17	4	0.17
5	0.23	6	0.23
7	0.23	8	0.23
9	0.14	10	0.14
11	0.14	12	0.14
13	0.08	14	0.08
15	0.08	16	0.08
17	0.05	18	0.05
19	0.05	20	0.05
21	0.05	22	0.04
23	0.04	24	0.04
25	0.04	26	0.06
27	0.06	28	0.06
29	0.06	30	0.04
31	0.04	32	0.04

33	0.04	34	0.04
35	0.04	36	0.04
37	0.04	38	0.15
39	0.15	40	0.15
41	0.15	42	0.15

Step 8: Calculating the Final Score of the documents. Final Score is calculated by averaging all the above scores.

Table 3.15: Calculation of final score of each sentence in Document 1

Sentence no	Final Score	Sentence no	Final Score
1	1.24979741095	2	0.876012855246
3	1.11777907914	4	0.514451915296
5	0.538775904842	6	0.510287449905
7	0.498689656278	8	0.638953854576
9	0.546458262543	10	0.596136705667
11	0.58525296625	12	0.846203871997
13	0.50874759976	14	0.32736008462
15	0.3513781339	16	0.388715107576
17	0.665832996266	18	0.481702939947
19	1.03188575953	20	0.58272572207
21	0.591351190449	22	0.406467390162

23	1.18285189422	24	0.789237331954
25	0.487582309144	26	0.347437594138
27	0.762138933602		

Table 3.16: Calculation of final score of each sentence in Document 2

Sentence no	Final Score	Sentence no	Final Score
1	0.85240009963	2	1.32435517633
3	1.18642186662	4	0.443494936319
5	0.949859147785	6	1.48886825273
7	0.766394349828	8	1.62795743968
9	0.913482735916	10	1.11514953461
11	0.564992591695	12	0.672611151751
13	0.445120233434	14	0.619031884605
15	0.985087172369	16	1.04574290542
17	2.08434584481	18	0.468702330329
19	1.63915831675	20	0.57242294789
21	0.825597752269	22	0.821156386102
23	0.440873112936	24	0.716423627485
25	0.532526776422	26	0.944536896838
27	1.60143268329	28	1.12764695591
29	1.3336555155	30	1.04128361783

31	0.996855801627	32	0.714907237976
33	1.11893954085	34	1.27188474466
35	0.911518414854	36	0.582429801205
37	1.29948999881	38	0.517511024299
39	1.06334603114	40	1.26958822577
41	0.758617468369	42	0.646279679512

Chapter 4
Experimental Analysis and Results

4.1 SOFTWARE REQUIREMENTS:

Programming Language:	Python
Operating System:	Windows, Linux
Front End:	HTML
Tool:	Pycharm IDE

4.1.1. INTRODUCTION TO 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 It provides constructs that enable clear programming on both small and large scales.

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 interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation

4.1.1.1. Python Library

Python's large standard library, commonly cited as one of its greatest strengths, provides tools suited to many tasks. For Internet-facing applications, many standard formats and protocols such as MIME and HTTP are supported. It includes modules for creating graphical user interfaces, connecting to relational databases, generating pseudorandom numbers, arithmetic with arbitrary precision decimals, manipulating regular expressions, and unit testing

Some parts of the standard library are covered by specifications (for example, the Web Server Gateway Interface (WSGI) implementation `wsgiref` follows PEP 333[90]), but

most modules are not. They are specified by their code, internal documentation, and test suites (if supplied). However, because most of the standard library is cross-platform Python code, only a few modules need altering or rewriting for variant implementations. The Python Package Index (PyPI),the official repository for third-party Python software, contains over 130,000 packages with a wide range of functionality, including:

- Graphical user interfaces
- Web frameworks
- Multimedia
- Databases
- Networking
- Test frameworks
- Automation
- Web scraping[92]
- Documentation
- System administration
- Scientific computing
- Text processing
- Image processing

4.1.1.2. Python Packages

1. NLTK: NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

2. Sckit-Learn: Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, *k* HYPERLINK "https:#en.wikipedia.org/wiki/K-means_clustering "HYPERLINK "https:#en.wikipedia.org/wiki/K-means_clustering" HYPERLINK

"https:#en.wikipedia.org/wiki/K-means_clustering"-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

4.1.1.3. Advantages Of Python

The Python language has diversified application in the software development companies such as in gaming, web frameworks and applications, language development, prototyping, graphic design applications, etc. This provides the language a higher plethora over other programming languages used in the industry.

- Extensive Support Libraries

It provides large standard libraries that include the areas like string operations, Internet, web service tools, operating system interfaces and protocols. Most of the highly used programming tasks are already scripted into it that limits the length of the codes to be written in Python.

- Integration Feature

Python integrates the Enterprise Application Integration that makes it easy to develop Web services by invoking COM or COBRA components. It has powerful control capabilities as it calls directly through C, C++ or Java via Jython. Python also processes XML and other markup languages as it can run on all modern operating systems through same byte code.

- Improved Programmer’s Productivity

The language has extensive support libraries and clean object-oriented designs that increase two to ten fold of programmer’s productivity while using the languages like Java, VB, Perl, C, C++ and C#.

- Productivity

With its strong process integration features, unit testing framework and enhanced control capabilities contribute towards the increased speed for most applications and productivity of applications. It is a great option for building scalable multi-protocol network applications.

4.2 HARDWARE REQUIREMENTS:

Processor: Intel Multi Core processor

RAM: 2 GB or above

Hard disk: 100 GB or above

4.2.1 User Interface:

The work of the user is to give a sentence as an input.

4.2.2 Hardware Interface:

MONITOR: the outputs are displayed on the monitor screen.

4.2.3 Software Interface:

Python is a programming language which supports machine learning algorithms for predicting accuracy and defining the sentiment of review.

4.3 Sample Code

```
# summarize.py
#Importing different Libraries in Python

from __future__  import print_function
from collections import Counter
from sklearn.feature_extraction.text import TfidfVectorizer
from nltk.corpus import reuters
from nltk.stem.snowball import SnowballStemmer
from nltk.corpus import stopwords
import nltk.data
import math
import re


DOC_ROOT = 'docs/'
DEBUG = False
SUMMARY_LENGTH = 5
stop_words = stopwords.words('english')
ideal_sent_length = 20.0
stemmer = SnowballStemmer("english")
```

```
class Summarizer3():

    def __init__(self, articles,compression):
        self._articles = []
        self.compression_prcnt=compression
        for doc in articles:
            with open(DOC_ROOT + doc) as f:
                headline = f.readline()
                url = f.readline()
                f.readline()
                body = f.read().replace('\n', ' ')
                if not self.valid_input(headline, body):
                    self._articles.append((None, None))
                    continue
                self._articles.append((headline, body))

    #checks Whether the Input is Valid or not


    def valid_input(self, headline, article_text):
        return headline != '' and article_text != ''

    #Tokenization and Stemming


    def tokenize_and_stem(self, text):
        tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sent)]
        filtered = []



        for token in tokens:
            if re.search('[a-zA-Z]', token):
                filtered.append(token)
        stems = [stemmer.stem(t) for t in filtered]
        return stems

    #Calculating Sentence Score
def score(self, article):

        headline = article[0]
        sentences = self.split_into_sentences(article[1])
        print("dividing the text document into sentences")
        print(sentences)
```

```
    frequency_scores = self.frequency_scores(article[1])
    j=1
    r=[]
    for i, s in enumerate(sentences):
        headline_score = self.headline_score(headline, s, j) * 1.5
        length_score = self.length_score(self.split_into_words(s)) * 1.0
        print("length score for sent %d" % j)
        print(length_score)
        position_score = self.position_score(float(i+1), len(sentences)) * 1.0
        print("position score for sent %d" % j)



        print(position_score)
        frequency_score = frequency_scores[i] * 4
        print("frequency score for sent %d" % j)
        print(frequency_score/4)
        score = (headline_score + frequency_score + length_score + position_score) / 4.0
        r.append([score,j])
        print("final score for sent %d" % j)
        print(j,score,sep = '  -  ')
        j=j+1
        self._scores[s] = score
    n=sorted(r,reverse=True)
    print(n)
```

#Generating the Summary

```
  def generate_summaries(self):
        total_num_sentences = 0
    for article in self._articles:


        total_num_sentences += len(self.split_into_sentences(article[1]))
    print("total sent %d" % total_num_sentences)


    SUMMARY_LENGTH=((self.compression_prcnt*total_num_sentences)/100)
    print("summary length %d" % SUMMARY_LENGTH)


    if total_num_sentences <= SUMMARY_LENGTH:
        return [x[1] for x in self._articles]


    self.build_TFIDF_model()
```

```
        self._scores = Counter()
        for article in self._articles:
            self.score(article)


        highest_scoring = self._scores.most_common(SUMMARY_LENGTH)
        #if DEBUG:
         #   print(highest_scoring)


        #print("## Headlines: ")
        headlines="## Headlines: \n"
        for article in self._articles:
            #print("- " + article[0])
         headlines+="- " + article[0]+"\n"



        return headlines,' '.join([sent[0] for sent in highest_scoring])


         def split_into_words(self, text):
        """ Split a sentence string into an array of words """
        try:
            text = re.sub(r'[^\w ]', '', text) # remove non-words
            return [w.strip('.').lower() for w in text.split()]
        except TypeError:
            return None


    def remove_smart_quotes(self, text):


        return text.decode('utf-8').strip().replace(u"\u201c","").replace(
u"\u2014","").replace(u"\u201d", "").replace(u"\u2019", "").replace(u"\u2018", "").replace(
u"\u20ac","")



    def split_into_sentences(self, text):
        tok = nltk.data.load('tokenizers/punkt/english.pickle')
        sentences = tok.tokenize(self.remove_smart_quotes(text))
        sentences = [sent.replace('\n', '') for sent in sentences if len(sent) > 10]
        return sentences



  #Calculating Headline score
```

```
  def headline_score(self, headline, sentence, num):

    title_stems = [stemmer.stem(w) for w in headline if w not in stop_words]
    sentence_stems = [stemmer.stem(w) for w in sentence if w not in stop_words]
    count = 0.0
    print("headline score of sent %d" % num)
    for word in sentence_stems:
       if word in title_stems:
          count += 1.0
    score = count / len(title_stems)
    print(score)
    return score
```

#Calculating Length Score

```
def length_score(self, sentence):

    len_diff = math.fabs(ideal_sent_length - len(sentence))
    return len_diff / ideal_sent_length
```

#Calculating Position Score

```
  def position_score(self, i, size)

    relative_position = i / size
    if 0 < relative_position <= 0.1:
       return 0.17
    elif 0.1 < relative_position <= 0.2:
       return 0.23
    elif 0.2 < relative_position <= 0.3:
       return 0.14
    elif 0.3 < relative_position <= 0.4:
       return 0.08
    elif 0.4 < relative_position <= 0.5:
       return 0.05
    elif 0.5 < relative_position <= 0.6:
       return 0.04
    elif 0.6 < relative_position <= 0.7:
       return 0.06
    elif 0.7 < relative_position <= 0.8:
       return 0.04
```

```
        elif 0.8 < relative_position <= 0.9:
            return 0.04
        elif 0.9 < relative_position <= 1.0:
            return 0.15
        else:
            return 0
```

#Calculating TF-IDF

```
    def build_TFIDF_model(self):

        token_dict = {}
        for article in reuters.fileids():
            token_dict[article] = reuters.raw(article)



        self._tfidf = TfidfVectorizer(tokenizer=self.tokenize_and_stem, stop_words='english', de-
code_error='ignore')
        tdm = self._tfidf.fit_transform(token_dict.values())  # Term-document matrix



    def frequency_scores(self, article_text):

        response = self._tfidf.transform([article_text])
feature_names = self._tfidf.get_feature_names() # stemmed words

        word_prob = {}  # TF-IDF individual word probabilities
        for col in response.nonzero()[1]:
            word_prob[feature_names[col]] = response[0, col]
        if DEBUG:
            print(word_prob)


        sent_scores = []
        print("words remained after performing stopword removal")
        print(word_prob)
        m=1
        for sentence in self.split_into_sentences(article_text):
            score = 0
            sent_tokens = self.tokenize_and_stem(sentence)
            print("after tokenization and stemming for sent %d" % m)
            print(sent_tokens)
            for token in (t for t in sent_tokens if t in word_prob):
```

```
            score += word_prob[token]
            print(token,word_prob[token],sep = ' - ')
        print("sentence score for  sentence %d before dividing with sent length" % m)
        print(m,score,sep = ' - ')
        print("sent lenght %d" % len(sent_tokens))
        m=m+1


    # Normalize score by length of sentence


        sent_scores.append(score / len(sent_tokens))
    print("frequency score obtained after dividing score with length of sentence")
    p=1
    for n in sent_scores:
        print(p,n,sep = '   - ')
        p=p+1
    return sent_scores
```

4.4) Screenshots:

4.4.1 GUI(graphical user interface)

Initially the documents which user wants to summarize have to be uploaded using GUI. The compression ratio has to be set and then click on execute to summarize the documents selected.

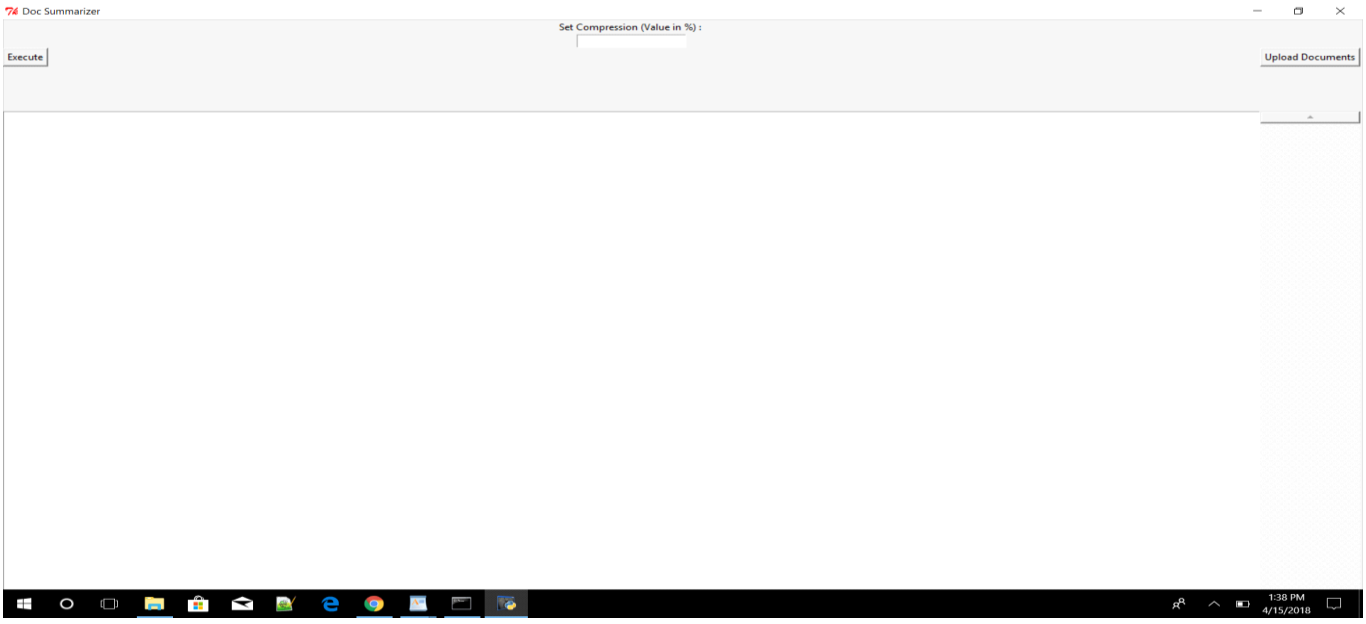

Figure 4.1: GUI

4.4.2) Input Documents

Two documents that are selected for summarization are shown below.

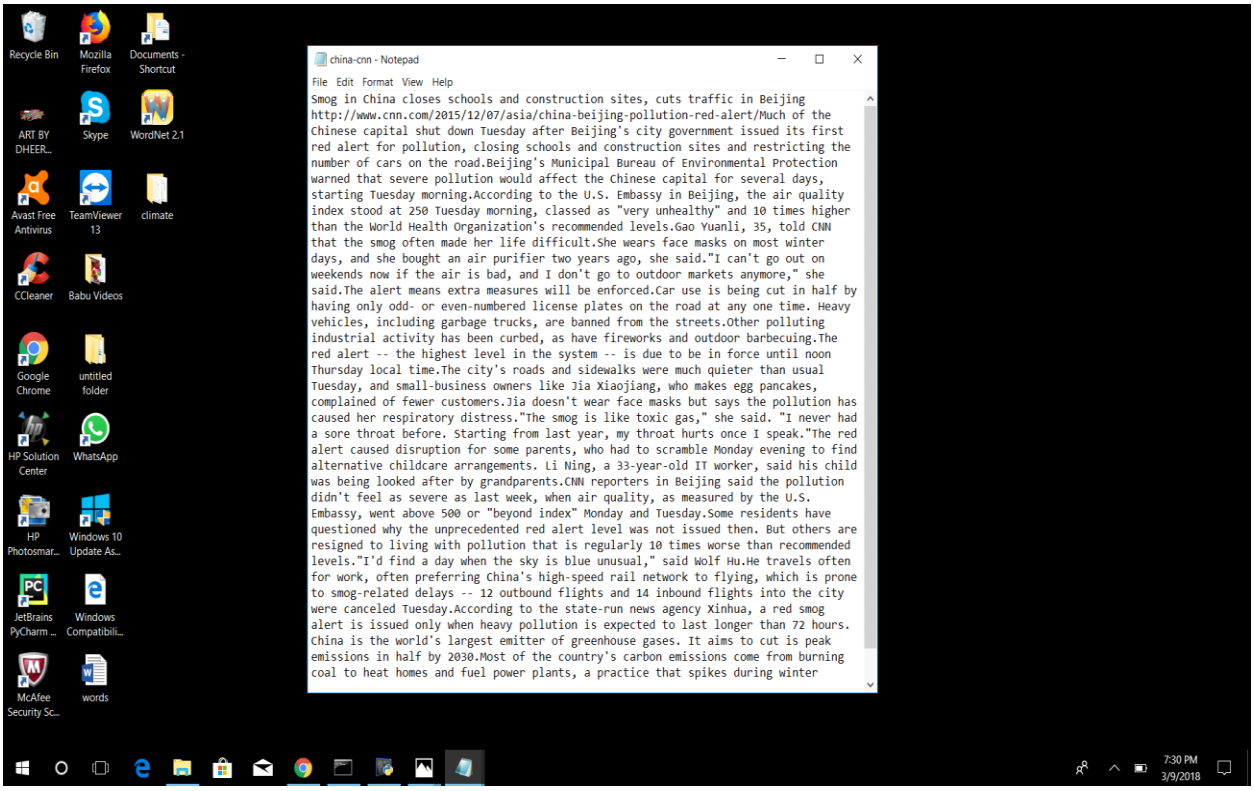

Figure 4.2:Input Document 1

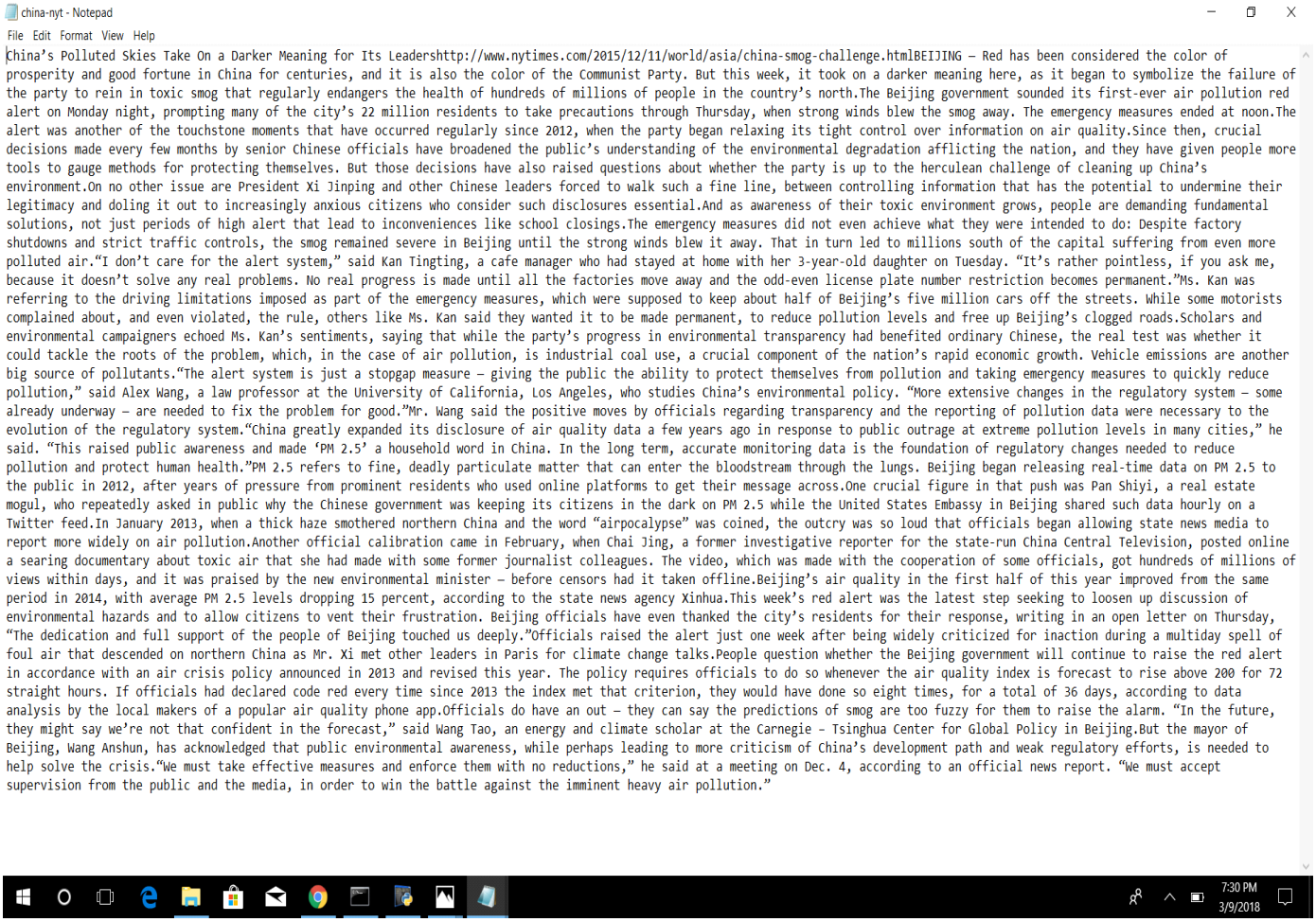

Figure 4.3:Input Document 2

4.4.3) Output Screenshots:

The compression ratio is set to 10 and then the output obtained after summarization is shown below.

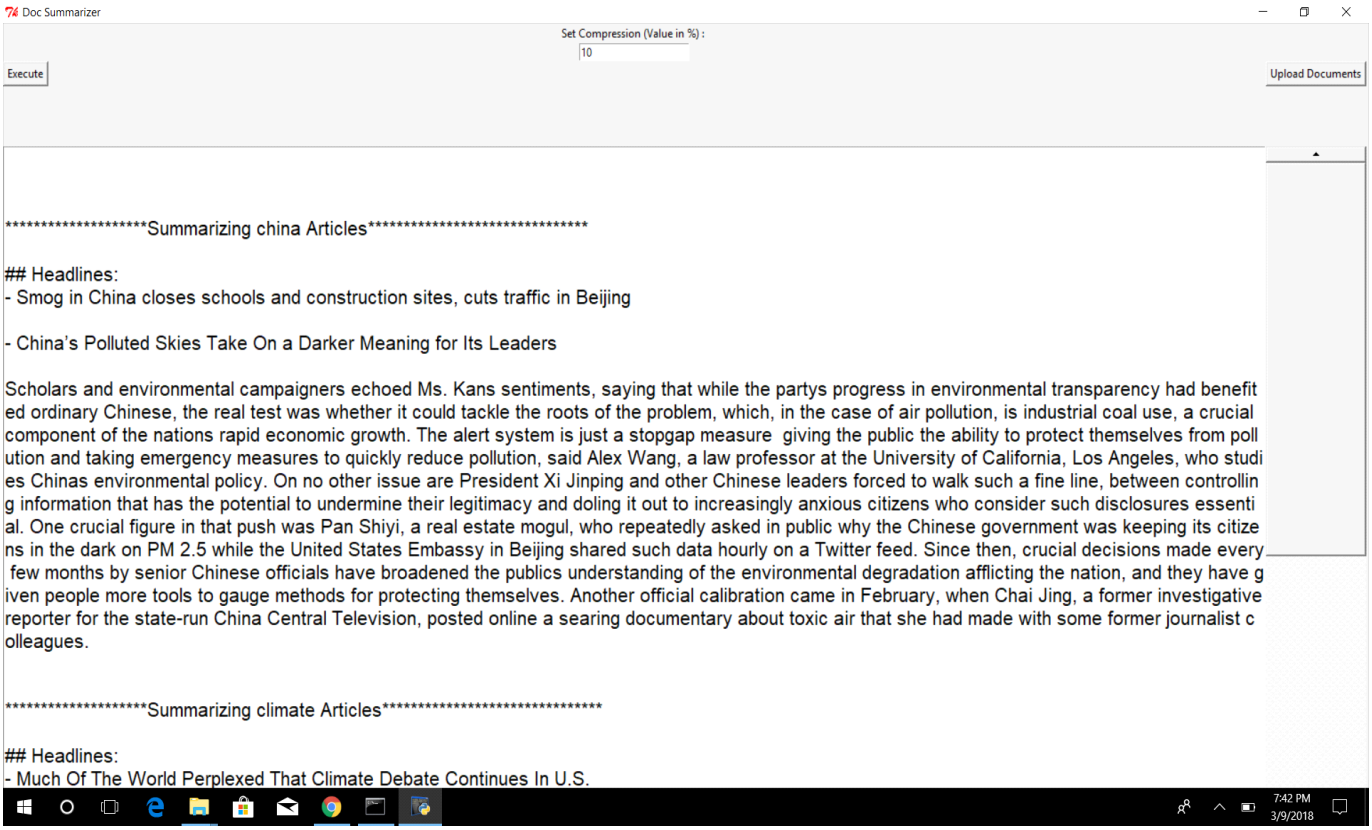

Figure 4.4: Output when compression ratio is 10%

The compression ratio is set to 20 and then the output obtained after summarization is shown below.

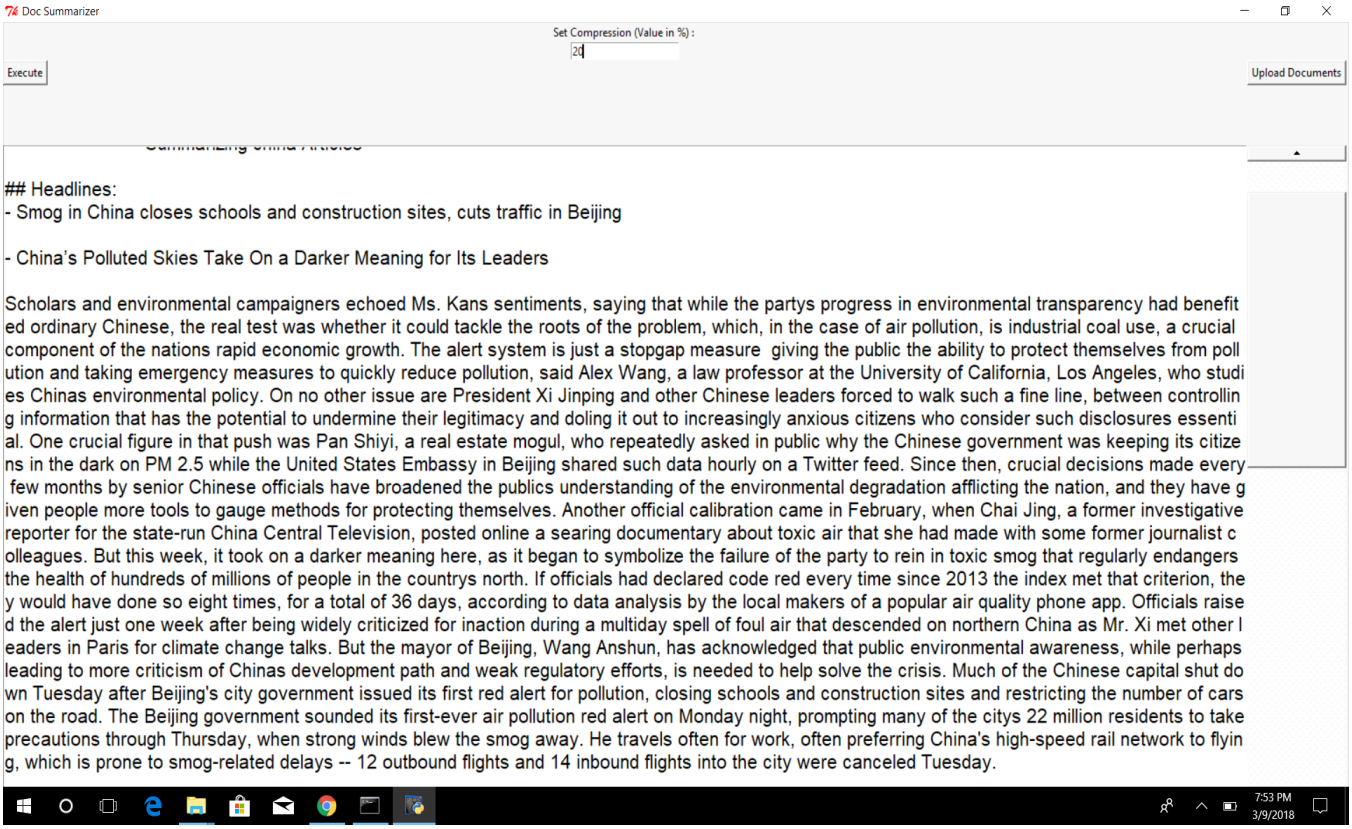

Figure 4.5: Output when compression ratio is 20%

The compression ratio is set to 40 and then the output obtained after summarization is shown below.

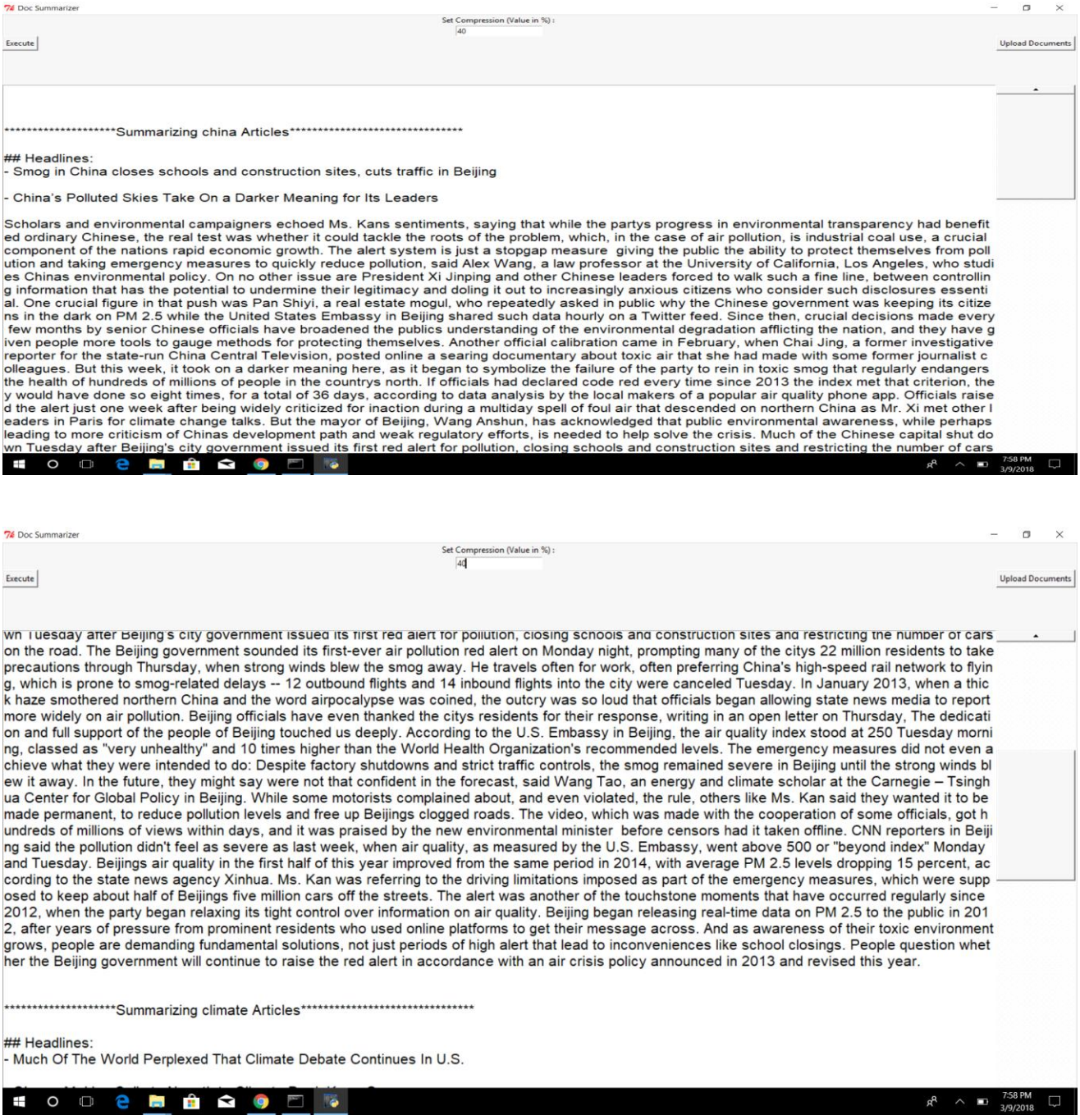

Figure 4.6, 4.7: Output when compression ratio is 40%

4.5 Evaluation Measures:

Precision:

Precision is well-suited to evaluating problems where the goal is to find a set of items from a larger set of items. In NLP, this can correspond to finding certain linguistic phenomena in a corpus.

Precision represents the proportion of items or entities that the system returns which are accurately correct. It rewards careful selection, and punishes over-zealous systems that return too many results: to achieve high precision, one should discard anything that might not be correct.

$$P = \frac{|\text{truepositives}|}{(|\text{truepositives}|+|\text{falsepositives}|)} \qquad \text{---------(4.1)}$$

Recall:

Recall is another important evaluating measure to express the goodness of the Natural Language Processing System. Recall indicates how much of all items that should have been found, were found. This metric rewards comprehensiveness: to achieve high recall, it is better to include entities that one is uncertain about. False negatives – missed entities – lead to low recall. It balances out precision.

$$R = \frac{|\text{TruePositives}|}{(|\text{TruePositives}|+|\text{falsenegatives}|)} \qquad \text{---------(4.2)}$$

Precision and Recall are used to calculate the efficiency of the proposed system.

Table 4.1 Precision and Recall Values for different documents

Document No.	Precision	Recall
1	0.7923	0.3913
2	0.8214	0.3636
3	0.8333	0.2857
4	0.7667	0.3448
Average	0.8034	0.3463

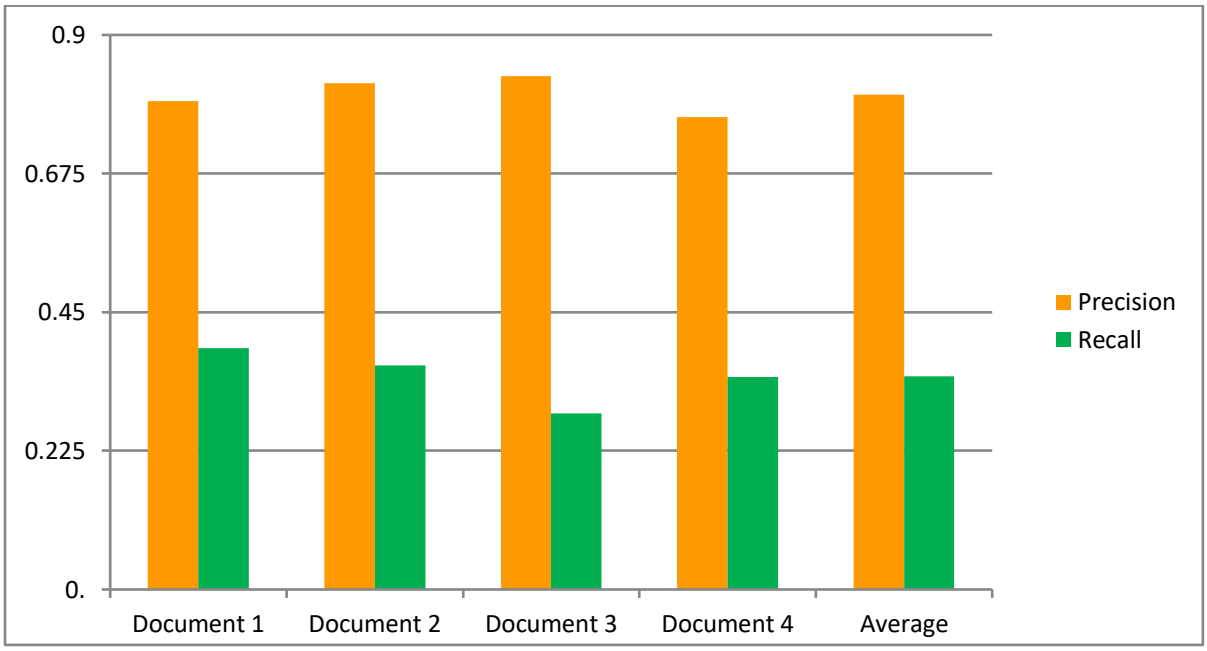

Figure 4.8 Bar-Graph showing Precision and Recall for different Documents

The above graph shows the Precision and Recall calculated for diffeent documents. Documents are summarized mannually and then compared with the system generated summaries. The average efficiency achieved in this project is 80.34%

Chapter 5
Conclusion and Future Scope

5.1 Conclusion:

In this project the proposed system was implemented for the multiple document summarization using algorithms of Feature extraction, TF-IDF, Position Score, Length Score, Headline Score. The proposed system can be used in educational institutions for students and lecturers in preparing their notes. Using this method, System gives the most important sentences in the article. Efficient feature extraction methods have been selected for this process. This technique can be applied for any text format. However, this cannot be applied for equations.

Each sentence is given the total score based on different scores obtained for an individual sentence. Different feature extraction methods like frequency score, position score, length score, headline score have been used. The number of lines in the output is determined by the compression ratio. To check the performance of existing system, recall and precision are used as evaluation measures.

5.2 Future Scope:

In the proposed system, synonyms for the word are not considered. If wordnet is used and probabilities are given according to the synonyms, the performance of the system can be increased. This system is not applicable to equations. Hence it can be extended to work even with equations.

REFERENCES

1. Daya C. Wimalasuriya, Dejing Dou,"Topic-Ontology-based information extraction: An introduction and a survey of current approaches.,First published March 19,2010

2. Eiman Tamah Al-Shammari, "Topic-Towards An ErrorFree Stemming",in proceedings of ADIS European Conference Data Mining 2008, pp. 160-163.

3. Gupta, V. K.,&Siddiqui,T.J., "Topic-Multidocument summarization using sentence clustering. In Intelligent Human Computer Interaction (IHCI), 2012 4th International Conference on (pp. 1-5).IEEE.".

4. Ho Chung Wu,Robert Wing Pong Luk,Kam Fai Wong. "Topic-Interpreting TF-IDF term weights as making relevance decisions.

5. HongyanLill et al."Topic-Multi-document Summarization based on Hierarchical Topic Model,HongyanLill, pp no 88-91".

6. HUANG Cheng-Hui,YIN Jian,HOU Fang,"Topic-Text Similarity Measurement Combining Word Semantic Information with TF-IDF Method".

7. Int.J.Advance.Soft Comput. Appl ., "Topic-Extraction Based Multi Document Summarization using Single Document Summary Cluster", Vol. 2, No. 1, March 2010.

8. Jyoti Singh, Jayshree Tembhare "Topic-A Review on Data Merging Techniques & Multi-Document Summarization", Volume 7 Issue no.3.

9. Krovetz Robert. "Topic-Viewing morphology as an inference process". Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval. 1993, 191-202.

10. Liu.N.Tang,WangH.W,Xiao.P.,"Topic-Sensitive Multi-document Summarization Algorithm. In Parallel Architectures, Algorithms and Programming (PAAP)", 2014 Sixth International Symposium on (pp. 69-74). IEEE.

11. Md.Majharul Haque,Suraiya Pervin and Zerina Begum , "Topic-Towards Efficient model for Automatic Text Summarization".

12. Mehdi Allahyari, Ms.Anjali,Ganesh Jivani in int J.Comp.Tech.Appl."Topic-A Comparative Study of Stemming Algorithms",Volume-2(6),1920-1938.

13. Oren, Nir. (2002). "Topic-Reexamining tf.idf based information retrieval with Genetic Programming". In Proceedings of SAICSIT 2002, 1-10.

14. Porter M.F. "Topic-An algorithm for suffix stripping". Program. 1980; 14, 130-137.

15. Porter M.F. "Topic-Snowball: A language for stemming algorithms". 2001.

16. Prasenjit Majumder,Mandar Mitra,Swapan K. Parui,Gobinda Kole,Pabitra Mitra and Kalyankumar Datta."Topic-YASS: Yet another suffix stripper". ACM Transactions on Information Systems. Volume 25, Issue 4. 2007, Article No. 18.

APPENDIX-A
List of stop words

a	about	above	after	again
against	all	am	an	and
any	are	aren't	as	at
be	because	been	before	being
below	between	both	but	by
can't	cannot	could	couldn't	did
didn't	do	does	doesn't	doing
don't	down	during	each	few
for	From	further	had	hadn't
has	hasn't	have	haven't	having
he	he'd	he'll	he's	her
here	here's	hers	herself	him
himself	his	how	how's	i
i'd	i'll	i'm	i've	if
in	into	is	isn't	It
it's	its	itself	Let's	me
more	most	mustn't	my	myself
no	nor	not	of	off
on	once	only	or	other
ought	our	ours	ourselves	out
over	own	same	shan't	She

she'd	she'll	she's	should	shouldn't
so	some	such	than	that
that's	the	their	theirs	Them
themselves	then	there	there's	These
they	they'd	they'll	they're	they've
this	those	through	To	Too
under	until	up	very	was
wasn't	we	we'd	we'll	we're
we've	were	weren't	what	what's
when	when's	where	where's	which
while	who	who's	whom	why
why's	with	won't	would	wouldn't
you	you'd	you'll	you're	you've
your	yours	Yourself	yourselves	----------

APPENDIX B:

List of Abbreviations

SNO	SHORTCUT	ABBREVIATION
1	NLP	Natural language processing
2	NLTK	Natural Language Toolkit
3	TF IDF	Term Frequency -Inverse Domain Frequency
4	SVM	Support Vector Machine
5	SVD	Singular-Value Decomposition
6	DUC	Document Understanding Conferences
7	ROGUE	Regional OnBase Group of User Experts
8	TAC	Text Analysis Conference
9	MEAD	Maintenance Engineering Analysis Data
10	ASCII	American Standard Code for Information Interchange
11	IDE	integrated development environment
12	MIME	Multipurpose Internet Mail Extensions
13	HTTP	Hyper Text Transfer Protocol
14	WSGI	Web Server Gateway Interface
15	DBSCAN	Density-Based Spatial Clustering of Application with Noise
16	GUI	Graphical User Interface