Chapter 1

INTRODUCTION

1.1 INTRODUCTION TO PROJECT

Document Summarization is the technique by which the huge parts of content are retrieved. The Document Summarization plays out the summarization task by unsupervised learning system. The significance of a sentence in info content is assessed by the assistance of Simplified Lesk calculation. As an online semantic lexicon WordNet is utilized. Word Sense Disambiguation (WSD) is a critical and testing system in the territory of characteristic dialect handling (NLP). A specific word may have distinctive significance in various setting. So, the principle task of word sense disambiguation is to decide the right feeling of a word utilized as a part of a specific setting. To begin with, Document Summarization assesses the weights of the considerable number of sentences of a content independently utilizing the Simplified Lesk calculation and orchestrates them in diminishing request as indicated by their weights. Next, as indicated by the given level of rundown, a specific number of sentences are chosen from that requested rundown.

Hybrid Document Summarization, is the plan to get an important data from a huge amount of information. The amount of data accessible on internet is increasing every day so it turns space and time expanding matter to deal with such huge amount of information. So, managing that large amount of data is makes a major problem in different and real data taking care of uses. The Hybrid Document Summarization undertaking makes the users simpler for various Natural Language applications, like, Data Recovery, Question Answering or content decreasing etc. Hybrid Document Summarization assumes an inescapable part by creating significant and particular data from a lot of information.

Filtering from heaps of reports can be troublesome and tedious. Without a summary or rundown, it can take minutes just to make sense of what the people will discuss in a paper or report. So, the Hybrid Document Summarization that concentrates a sentence from a content record, figures out which are the most imperative, and returns them in a readable and organized way. Hybrid Document

Summarization is a piece of the field natural language processing, which is the manner by which the PCs can break down, and get importance from human dialect.

Hybrid Document Summarization that uses the classifier structure and its rundown modules to look over huge amount of reports and returns the sentences that are helpful for producing a summary. Programmed outline of content works by taking the overlapping sentences and synonymous or sense from wordnet most overlapping sentences are considered as high score words. The higher recurrence words are considering most worth. And the top most worth words and are taking from the content and sorted according to its recurrence and generate a summary.

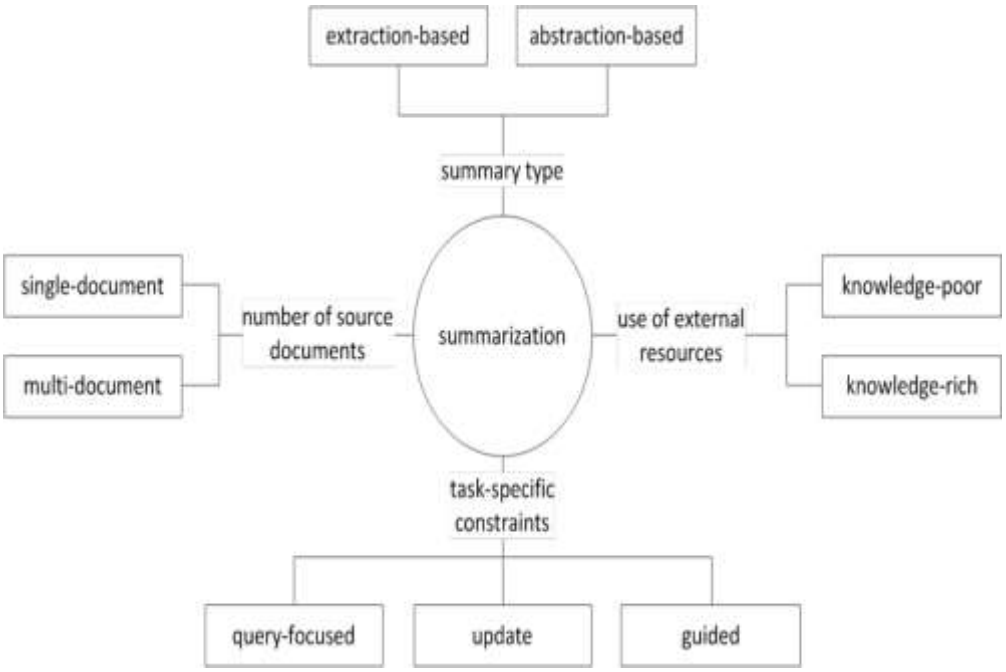


Figure 1.1: Classification of summarization tasks

1.2 OBJECTIVE AND SCOPE OF PROJECT

In the Hybrid Document Summarization, we are using a solitary or single input content is going to outlined by the given rate of summarization utilizing unsupervised learning. In any case, the streamlined lesk's computation is associated with each of the sentences to find the guarantees of each sentence. After that, sentences with induced weights are composed in sliding solicitation concerning

their weights. Presently as per a particular rate of summarization at a specific occurrence, certain quantities of sentences are chosen as an outline. The proposed computations, abridges solitary or single report content utilizing unsupervised learning approach. Here, the heaviness of every sentence in a substance is resolved using streamlined Lesk's computation and wordnet. After that, summarization procedure is performed as indicated by the given rate of synopsis. In which, we are taking solitary info content and display summarization as yield. First info content is passed, to the lesk computation and wordnet, where the weights of each sentences of the content are inferred utilizing and semantic investigation of the concentrates are performed. Next, weight doled out sentences is passed to derive the final summary according to the percentage of synopsis, where the last abridged outcome is assessed as and showed.

1.3 IMPORTANCE OF THE PROJECT

In this project we have implemented three different techniques for extractive based text summarization. Mainly the approaches used are different in terms of features used and machine learning model. In the first approach, we used sentence position, positive keywords in sentence, negative keywords in sentence, sentence centrality, sentence inclusion of name entity, sentence inclusion of numerical data, and sentence relative length [2]. The model is trained using Genetic Algorithms [2]. In the second approach, we chose to work on documents which are well structured and in which sentences are less connected. For summarization using this technique, we have used Wikipedia articles, as they provide structured content, and sentences are less connected. In this approach google search, wordnet word similarity [3], tf-idf, and sentence position features are used. In the third approach, continuous bag of words and skip-gram architecture are implemented using Word2Vec toolkit [4] and neural networks are used to train the summarizer. Reading the whole document, dismembering it and isolating the critical thoughts from the crude content require some serious energy and exertion. Perusing a document of 600 words can take no less than 10 minutes. Programmed outline programming condense writings of 500-5000 words in a brief instant. This enables the client to peruse less information yet get the most essential data and make strong conclusion. It reduces the human effort while creating a synopsis. A few vital products compress records as well as website pages. The persons quickly determine which points are imported for reading.

Chapter 2

LITERATURE SURVEY

Document Summarization that uses the classifier structure and its rundown modules to look over huge amount of reports and returns the sentences that are helpful for producing a summary are the ones that we have . Programmed outline of content works by taking the overlapping sentences and synonymous or sense from wordnet most overlapping sentences are considered as high score words. The higher recurrence words are considering most worth. And the top most worth words and are taking from the content and sorted according to its recurrence and generate a summary.

Abdel Fattah, Mohamed *et al.* [2] has proposed a simple approach for text summarization. They have considered features like position, length, name entities, numerical data, bushy paths, vocabulary overlaps etc. to generate summary. In this approach, sentences are modelled as vectors of features. Sentences are marked as correct if they are to be put in summary while are marked as incorrected if not. While making the final choice of sentences, each sentence is given a value in between 0 and 1 and using a machine learning model, the sentences are selected using those scores.

Various other papers have also been published which are very useful for a particular class of document. Kamal Sarkar *et al.* [5] is built for summarization of medical documents using machine learning approach. Various features are very common and specific to medical documents. It uses the concept of cue phases such that if a sentence contains n cue phases, it gets n as its score for the feature. It also uses position of cue phases in the document such that if it appears at the beginning or at the end of the sentence, it gets an additional 1 point. Acronyms are also used as a feature and sentences having these gets extra points. In some papers [6], sentences are also broken down by special cue makers and sentences are represented as a feature vector.

Ryang, D Seonggi *et al.* [7] proposed a method of automatic text summarization with reinforcement learning. Researches have also been done for summarization of Wikipedia articles. Hingu, D *et al.* [8] implemented various method for summering Wikipedia pages. In one of their methods, sentences containing citations are given higher weightage. In the other approach, the frequency of

words are adjusted based on the root form of the word. The words are stemmed with the objective to assign equal weightage to words with the same root word.

Edmundson (1969) [9] proposed an approach of extraction-based summarization using features like position, frequency of words, cue words and the *skeleton* of an article by manually assigning weight to each of these features. The system was tested using manually generated summaries of 400 technical documents. The results were good with 44% of summaries generated by it were matching the manual summaries

2.1 PHASES OF DOCUMENT SUMMARIZATION

There are basic three phases of document summarization as follows:

2.1.1 PRE-PROCESSING

Pre-processing is defined here as cleaning the data. For cleaning unwanted characters, symbols, extra spaces, hyperlinks etc are removed. The stop words like ‘a’, ‘the’, ’and’. Stemming is done where the words are reduced to their word root for example ‘playing’ would be reduced to ‘play’. Moreover the parsing of the above data is done where the words are bifurcated into nouns, adjectives, verbs etc. The pre-processing is done as per the requirement using one or combination of the above defined techniques.

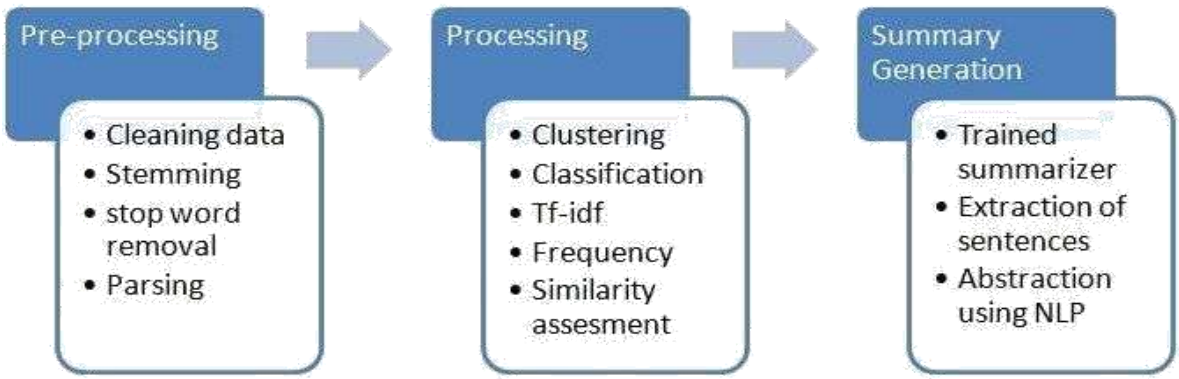


Fig 2.1 Phases of Summarization

This is the next phase of document summarization. After the Text data is cleaned and pre-processed, the processing techniques are applied in which the tf-idf scores, frequency of words are calculated.

The data is grouped on the basis of similarity, dissimilarity so that the summary generation can be made efficiently. For this different clustering techniques are used.

2.1.2 SUMMARY GENERATION

After the data is processed, the summary is to be generated based on the requirement. Extraction of sentences is done from the processed data and the summary is processed. The words to added to sentences, reducing the sentences based on score etc is done in this step to produce summary. The representation of the summary can be in the form of words, sentences, paragraphs, graphs etc. for the summary to be generated different summarizers are used.

2.2 AUTOMATIC KEYWORD EXTRACTION

On the premise of past work done towards automatic keyword extraction from the text for its summarization, extraction systems can be classified into four classes, namely, simple statistical approach, linguistics approach, machine learning approach, and hybrid approaches.

2.2.1 SIMPLE STATISTICAL APPROACH

These strategies are rough, simplistic and have a tendency to have no training sets. They concentrate on statistics got from non-linguistic features of the document, for example, the position of a word inside the document, the term frequency, and inverse document frequency. These insights are later used to build up a list of keywords. Cohen, utilized n-gram statistical data to discover the keyword inside the document automatically. Other techniques in- side this class incorporate

word frequency, term frequency (TF) or term frequency inverse document frequency (TF-IDF), word cooccurrences, and PAT-tree. The most essential of them is term frequency. In these strategies, the frequency of occurrence is the main criteria that choose whether a word is a keyword or not. It is extremely unrefined and tends to give very unseemly results. An improvement of this strategy is the TF-IDF, which also takes the frequency of occurrence of a word as the model to choose a keyword or not. Similarly, word co-occurrence methods manage statistical information about the number of times a word has happened and the number of times it has happened with another word. This statistical

information is then used to compute support and confidence of the words. Apriori technique is then used to infer the keywords.

2.2.2 LINGUISTICS APPROACH

This approach utilizes the linguistic features of the words for keyword detection and extraction in text documents. It incorporates the lexical analysis, syntactic analysis, discourse analysis, etc. The resources used for lexical analysis are an electronic dictionary, tree tagger, WordNet, engrams, POS pattern, etc. Similarly, noun phrase (NP), chunks (Parsing) are used as resources for syntactic analysis.

2.2.3 MACHINE LEARNING APPROACH

Keyword extraction can also be seen as a learning problem. This approach requires manually annotated training data and training models. Hidden Markov model, support vector machine (SVM) , naive Bayes (NB) , bagging , etc. are commonly used training models in these approaches. In the second phase, the document whose keywords are to be extracted is given as inputs to the model, which then extracts the keywords that best fit the model's training. One of the most famous algorithms in this approach is the keyword extraction algorithm (KEA). In this approach, the article is first converted into a graph where each word is treated as a node, and whenever two words appear in the same sentence, the nodes are connected with an edge for each time they appear together. Then the number of edges connecting the vertices are converted into scores and are clustered accordingly. The cluster heads are treated as keywords. Bayesian algorithms use the Bayes classifier to classify the word into two categories: keyword or not a keyword depending on how it is trained. GenEx is another tool in this approach.

2.2.4 HYBRID APPROACH

These approaches combine the above two methods or use heuristics, such as position, length, layout feature of the words, HTML tags around the words, etc. These algorithms are designed to take the best features from above mentioned approaches. Based on the classification we bring a consolidated summary of previous studies on automatic keyword extraction. It discusses the approaches that are used for keyword extraction and various domains of dataset in which experiments are performed.

2.3 DOCUMENT SUMMARIZATION PROCESS

Based on the literature, text summarization process can be characterized into five types, namely, based on the number of the document, based on summary usage, based on techniques, based on characteristics of summary as text and based on levels of linguistics process.

2.3.1 SINGLE DOCUMENT TEXT SUMMARIZATION

In single document text summarization, it takes a single document as an input to perform summarization and produce a single output document. Thomas designed a system for automatic keyword extraction for text summarization in single document e-Newspaper article. Marcu developed a discourse-based summarizer that determines adequacy for summarizing texts for discourse- based methods in the domain of single news articles.

2.3.2 MULTIPLE DOCUMENT TEXT SUMMARIZATION

In multiple documents text summarization, it takes numerous documents as an input to perform summarization and deliver a single output document. Mirroshandel presents two different algorithms towards temporal relation-based keyword extraction and text summarization in multi-document. The first algorithm was a weakly supervised machine learning approach for classification of temporal relations between events and the second algorithm was expectation maximization (EM) based unsupervised learning approach for temporal relation extraction. Min used the information which is common to document sets belonging to a common category to improve the quality of automatically extracted content in multi-document summaries.

2.3.3 QUERY-BASED TEXT SUMMARIZATION

In this summarization technique, a particular portion is utilized to extract the essential keyword from input document to make the summary of corresponding document. Fisher developed a query- based summarization system that uses a log-linear model to classify each word in a sentence. It exploits the property of sentence ranking methods in which they consider neural query ranking and query-focused ranking. Dong developed a query-based summarization that uses document ranking, time-sensitive queries and ranks recency sensitive queries as the features for text summarization. They

showed that CSI is a valuable metric to aid sentence selection in extractive summarization tasks. Marcu developed a discourse based extractive summarizer that uses the rhetorical parsing algorithm to determine discourse structure of the text of given input, determine partial ordering on the elementary and parenthetical units of the text.

2.3.4 EXTRACTIVE TEXT SUMMARIZATION

In this procedure, summarizer discovers more critical information (either words or sentences) from input document to make the summary of the corresponding document. In this process, it uses statistical and linguistic features of the sentences to decide the most relevant sentences in the given input document. Thomas designed a model based extractive summarizer using machine learning and simple statistical method for keyword extraction

from e-Newspaper summarize the text in multi-document. They used information which is common to document sets belonging to a common category as a feature and encapsulated the concept of category-specific importance (CSI). They showed that CSI is a valuable metric to aid sentence selection in extractive summarization tasks. Marcu developed a discourse based extractive summarizer that uses the rhetorical parsing algorithm to determine discourse structure of the text of given input, determine partial ordering on the elementary and parenthetical units of the text. Erkan developed an extractive summarization environment. It consists of three steps: feature extractor, the feature vector, and reranked.

Features are Centroid, Position, Length Cut-off, SimWithFirst, Lex PageRank, and QueryPhraseMatch. Alguliev developed an unsupervised learning based extractive summarizer that optimizes three properties: relevance, redundancy, and length. It split documents into sentences and select salient sentences from the document. Aramaki destined a supervised learning based extractive text summarizer that identifies the negative event and it also investigates what kind of information is helpful for negative event identification. An SVM classifier is used to distinguish negative events from other events.

2.3.5 ABSTRACTIVE TEXT SUMMARIZATION

In this procedure, a machine needs to comprehend the idea of all the input documents and then deliver summary with its particular sentences. It uses linguistic methods to examine and interpret the text and then to find the new concepts and expressions to best describe it by generating a new shorter text that conveys the most important information from the original text document. Brandow developed an abstractive summarization system that analyses the statistical corpus and extracts the signature words from the corpus. Then it assigns the weight for all the signature words. Based on the extracted signature words, they assign the weight to the sentences and select few top weighted sentences as the summary. Daume developed an abstractive summarization system that maps all the documents into database-like representation. Further, it classifies into four categories: a single person, single event, multiple event, and natural disaster. It generates a short headline using a set of predefined templates. It generates summaries by extracting sentences from the database.

2.3.6 SUPERVISED LEARNING BASED TEXT SUMMARIZATION

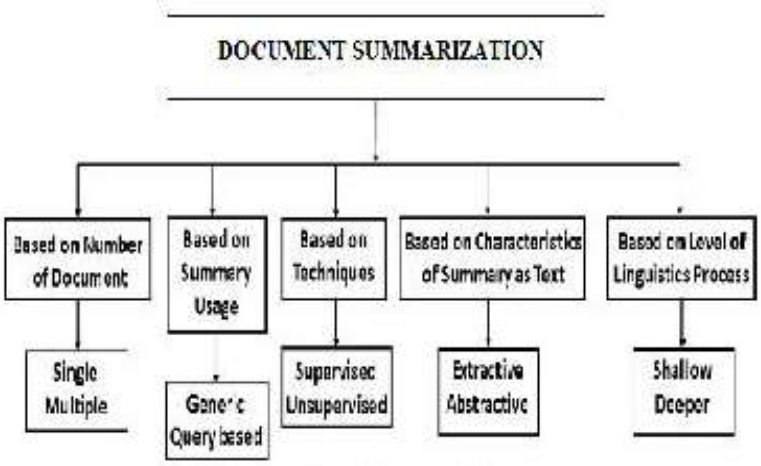


Fig 2.2 Text summarization process

This type of learning techniques used labelled dataset for training. Thomas designed a system for automatic keyword extraction for text summarization using hidden Markov model. The learning process was supervised, it used human annotated keyword set to train the model. used a set of labelled datasets to train the system for the classification of temporal relations between events. Destined a

supervised learning based extractive text summarizer that identifies the negative event. Article used freely available, open-source extractive summarization system, called SWING to and also investigates what kind of information is helpful for negative event identification. An SVM classifier is used to distinguish negative events from other events.

In this technique, there are no predefined guidelines available at the time of training. Mirroshandel. proposed a method for temporal relation extraction based on the Expectation-Maximization (EM) algorithm. Within EM, they used different techniques such as a greedy best-first search and integer linear programming for temporal inconsistency removal. The EM-based approach was a fully unsupervised temporal relation-based extraction for text summarization. Alguliev developed an unsupervised learning based extractive summarizer that optimizes three properties: relevance, redundancy, and length. It split documents into sentences select salient sentences from the document.

2.4 ISSUES AND CHALLENGES OCCURS IN TEXT SUMMARIZATION

In the area of text summarization, there are following research issues and challenges occurs during implementation.

2.4.1 RESEARCH ISSUES

- In the case of multi-document text summarization, several issues occur frequently while evaluation of summary such as redundancy, temporal dimension, co-reference or sentence ordering, etc. which makes very difficult to achieve quality summary. Some other issues occur such as grammaticality, cohesion, coherence which is harmful for summary.
- The quality of summaries are varying from system to system or person to person. Some person feels some set of sentences are important for summary, at the same time other person feel the other set of sentences are important for required summary.

2.4.2 IMPLEMENTATION CHALLENGES

- To get the quality summary, quality keywords are required for text summarization.
- There is no standard to identify quality keywords within or multiple documents. The extracted keywords are varying for applying different approaches of keyword extraction.
- Multi-lingual text summarization is another challenging task.

REF NO	TITLE	METHODLOGY	ALGORITHM	ACCURACY
1	A Neural Attention Model for Abstractive Sentence Summarization	local attention-based model	Beam Search	43.81%
2	Abstractive Text Summarization using Sequence-to-sequence RNN and Beyond	RNN	-	74.57%
3	Automatic Keyword Extraction for Text Summarization : survey	-	-	-
4	Summarization with Pointer-Generator Networks	Pointer generator network and CNN	-	-
5	Text Summarization Techniques: A Brief Survey	Frequency Driven Approaches	-	-
6	Ranking Sentences for Extractive Summarization with Reinforcement Learning	CNN/RNN	REINFORCE algorithm	37.98%
7	Neural Document Summarization by Jointly Learning to Score and Select Sentences	RNN	-	66.34%
8	Machine Learning Techniques for Document Summarization: A Survey	Talks about all the methods	-	-
9	Ranking with Recursive Neural Networks and Its Application to Multi-Document Summarization	R2N2/RNN	-	-
10	A Neural Attention Model for Sentence Summarization	local attention-based model	Beam Search	23.97%
11	A Review Paper on Text Summarization	Natural Language Processing	-	-
12	Automatic Text Summarization Using Natural Language Processing	Natural Language Processing	LESK Algorithm	-
13	A Novel Technique for Efficient Text Document Summarization as a Service	Natural Language Processing	LESK Algorithm	33%
14	Automatic Text Summarization and	Term frequency-based		

	its methods : A Review	method/ Graph based method/ Time based method/Clustering based method	-	-
15	A Review on Automatic Text Summarization Approaches	Frequency Based Approach/Term Frequency–Inverse Document	-	-

Fig 3.3 Comparison of Different Papers

Chapter 3

PREAMBLE

3.1 EXISTING SYSTEM

Currently, the number of documents retrieved by Web Search Engines is already beyond the capacity of human analysis due to the fact that hundreds of pages of search results are generated for most input queries. Thus, document retrieval is not sufficient and we need a second level of abstraction to reduce this huge amount of data - the ability of summarization. Hybrid Document Summarization condenses text contents into most important concepts and ideas under a particular context. This technology may be helpful to identify topics, categorize contents, and summarize documents. However, most previous work on Hybrid Document Summarization has emphasized on information abstraction and extraction. Some well-known approaches, like TF/IDF (Term Frequency/Inverse Document Frequency), which summarizes a text based on term frequency weight that is assigned to each term, neural network system for text summarization, statistical models, and so on, usually rank sentences and select sentences with higher ranking Score as the summary.

There are two properties of the summary that must be measured while evaluating summaries and summarization systems – the Compression Ratio, which is a measure of the length of the summary when compared to the original, and the Retention Ratio or Omission Ratio, which is a measure of how much of the document's central information is retained in the summary.

Semantic similarity is a concept frequently employed in determining the ranking of a term or sentence. A set of documents or terms within term lists are assigned a metric based on the likeness of their meaning/semantic content. Various semantic similarity techniques are available which can be used for measuring the semantic similarity between text documents. Semantic similarity methods are classified into four main categories, Edge Counting Methods that measure the similarity between two terms (concepts) as a function of the length of the path linking the terms and on the position of the terms in the taxonomy, Information Content Methods to measure the difference in information content of two terms as a function of their probability of occurrence in a corpus, Feature based Methods to

measure similarity between two terms as a function of their properties (e.g., their definitions) or based on their relationships to other similar terms in the taxonomy and Hybrid methods that combine the above three mentioned methods for calculating the semantic similarity.

3.2 PROBLEM STATEMENT

- Reading the whole document, dismembering it and isolating the critical thoughts from the crude content require some serious energy and exertion.
- Existing system increase the human effort while creating a synopsis. A few vital products compress records as well as website pages.
- The persons cannot quickly determine which points are imported for reading.

3.3 PROPOSED SYSTEM

In the Hybrid Document Summarization, we are using a solitary or single input content is going to outlined by the given rate of summarization utilizing unsupervised learning. In any case, the streamlined lesk's computation is associated with each of the sentences to find the guarantees of each sentence. After that, sentences with induced weights are composed in sliding solicitation concerning their weights. Presently as per a particular rate of summarization at a specific occurrence, certain quantities of sentences are chosen as an outline. The proposed computations, abridges solitary or single report content utilizing unsupervised learning approach. Here, the heaviness of every sentence in a substance is resolved using streamlined Lesk's computation and wordnet. After that, summarization procedure is performed as indicated by the given rate of synopsis. In which, we are taking solitary info content and display summarization as yield. First info content is passed, to the lesk computation and wordnet, where the weights of each sentences of the content are inferred utilizing and semantic investigation of the concentrates are performed. Next, weight doled out sentences is passed to derive the final summary according to the percentage of synopsis, where the last abridged outcome is assessed as and showed.

Where, input document will be in the form of a word document file or a pdf file. The pre-processing includes the data cleaning and data abstraction. The data will be the input to the lesk algorithm with the weights given to the words. Wordnet acts as a dictionary for comparing the importance of the

word given is the input. Once the data is processed in the lesk algorithm it gives the output values which will be further converted into the summarized document format.

3.4 ADVANTAGES:

- Reading the whole document, dismembering it and isolating the critical thoughts from the crude content require some serious energy and exertion. Perusing a document of 600 words can take no less than 10 minutes. Programmed outline programming condense writings of 500-5000 words in a brief instant. This enables the client to peruse less information yet get the most essential data and make strong conclusion.
- It reduces the human effort while creating a synopsis. A few vital products compress records as well as website pages. The persons quickly determine which points are imported for reading.

CHAPTER 4

SYSTEM REQUIREMENTS

System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently.

4.1 HARDWARE REQUIREMENTS

System	**:**	I5 PROCESSOR
Hard Disk	**:**	250 GB.
RAM	**:**	8 GB
GPU	**:**	4 GB

- Any desktop / Laptop system with above configuration or higher level

4.2 SOFTWARE REQUIREMENTS

Operating system	:	Windows 7/8/10
Coding Language	:	Python
Web Technology	:	Django
IDE	:	Sublime Text 3, Anaconda

4.3 FUNCTIONAL REQUIREMENTS

- Functional requirements specifies a function that a system or system component must be able to perform. It can be documented in various ways. The most common ones are written descriptions in documents, and use cases.
- A typical functional requirement will contain a unique name and number, a brief summary, and a rationale. This information is used to help the reader understand why the requirement is needed, and to track the requirement through the development of the system.

- Functional requirements is what a system is supposed to accomplish. It may be calculations, technical details, data manipulations, data processing etc.

4.4 NON-FUNCTIONAL REQUIREMENTS

- Non-functional requirements are any other requirement than functional requirements. This are the requirements that specifies criteria that can be used to judge the operation of a system, rather than specific behaviours. Non-functional requirements place restrictions on the product being developed, the development process, and specify external constraints that the product must meet.
- Non-functional requirements - can be divided into two main categories:

Execution qualities, such as security and usability, which are observable at run time.

Evolution qualities, such as testability, maintainability, extensibility and scalability, which are embodied in the static structure of the software system.

CHAPTER 5

SYSTEM ARCHITECTURE

A system architecture is an orderly group of interdependent components linked together according to a plan to achieve a specific objective. Its main characteristics are organization, interaction, interdependence, integration and a central objective.

Analysis is a detailed study of the various operations performed by a system and their relationships within and outside of the system. One aspect of analysis is defining the boundaries of the system and determining whether or not a candidate system should consider other related systems. During analysis data are collected on the available files decision points and transactions handled by the present system. This involves gathering information and using structured tools for analysis.

System analysis and design are the application of the system approach to problem solving generally using computers. To reconstruct a system the analyst must consider its elements output and inputs, processors, controls, feedback and environment.

5.1 LOGICAL DESIGN

Design for Frontends encompasses technical and non-technical activities. The look and feel of content is developed as part of graphic design; the aesthetic layout of the user interface is created as part of interface design; and the technical structure of the Frontend is modeled as part of architectural and navigational design.

It answers three primary questions for the end-user:

- Where am I? – The interface should (1) provide an indication of the Frontend has been accessed and (2) inform the user of her location in the content.
- What can I do now? – The interface should always help the user understand his current options-what functions are available, what links are live, what content is relevant.

- Where have I been; where am I going? – The interface must facilitate navigation. Hence it must provide a "map" of where the user has been and what paths may be taken to move elsewhere in the Frontend.

5.2 DESIGN GOALS

The following are the design goals that are applicable to virtually every Frontend regardless of application domain, size, or complexity.

- Simplicity
- Consistency
- Identity
- Visual appeal
- Compatibility.

Design leads to a model that contains the appropriate mix of aesthetics, content, and technology. The mix will vary depending upon the nature of the Frontend, and as a consequence the design activities that are emphasized will also vary.

5.2.1 THE ACTIVITIES OF THE DESIGN PROCESS:

- Interface design-describes the structure and organization of the user interface. Includes a representation of screen layout, a definition of the modes of interaction, and a description of navigation mechanisms. Interface Control mechanisms- to implement navigation options, the designer selects form one of a number of interaction mechanism. Interface Design work flow- the work flow begins with the identification of user, task, and environmental requirements. Once user tasks have been identified, user scenarios are created and analyzed to define a set of interface objects and actions.

- Aesthetic design-also called graphic design, describes the "look and feel" of the Frontend. Includes color schemes, geometric layout. Text size, font and placement, the use of graphics, and related aesthetic decisions.

- Content design-defines the layout, structure, and outline for all content that is presented as part of the Frontend. Establishes the relationships between content objects.
- Navigation design-represents the navigational flow between contents objects and for all Frontend functions.
- Architecture design-identifies the overall hypermedia structure for the Frontend. Architecture design is tied to the goals establish for a Frontend, the content to be presented, the users who will visit, and the navigation philosophy that has been established.
- Content architecture, focuses on the manner in which content objects and structured for presentation and navigation.
- Frontend architecture, addresses the manner in which the application is structure to manage user interaction, handle internal processing tasks, effect navigation, and present content. Frontend architecture is defined within the context of the development environment in which the application is to be implemented.

5.3 DATA FLOW DIAGRAM

A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system (usually an information system). The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

5.3.1 CONTEXT ANALYSIS

Context analysis is a method to analyze the environment in which a business operates. Environmental scanning mainly focuses on the macro environment of a business. But context analysis considers the entire environment of a business, its internal and external environment.

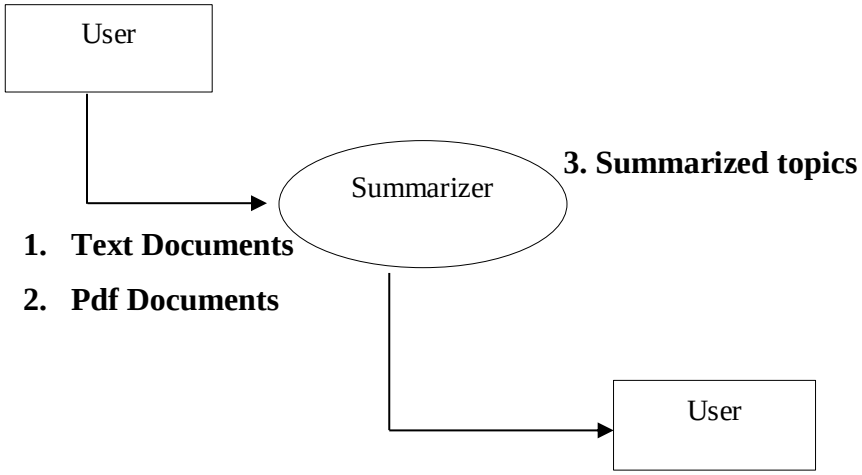


Figure 5.2: Context Analysis

5.3.2 SOCI RANK DATA PROCESS

Dataflow is the movement of data through a system comprised of software, hardware or a combination of both. Dataflow is often defined using a model or diagram in which the entire process of data movement is mapped as it passes from one component to the next within a program or a system, taking into consideration how it changes form during the process. The dataflow diagram is important in the architectural design of a system since it defines what kind of data is needed in order to start or complete a specific process. The process of summarization begins with processing of input document which is broken down into sentences and subsequently into words. Summarizer maintains a list of sentences of the document and each sentence is responsible to store the words contained in it. A UML class diagram

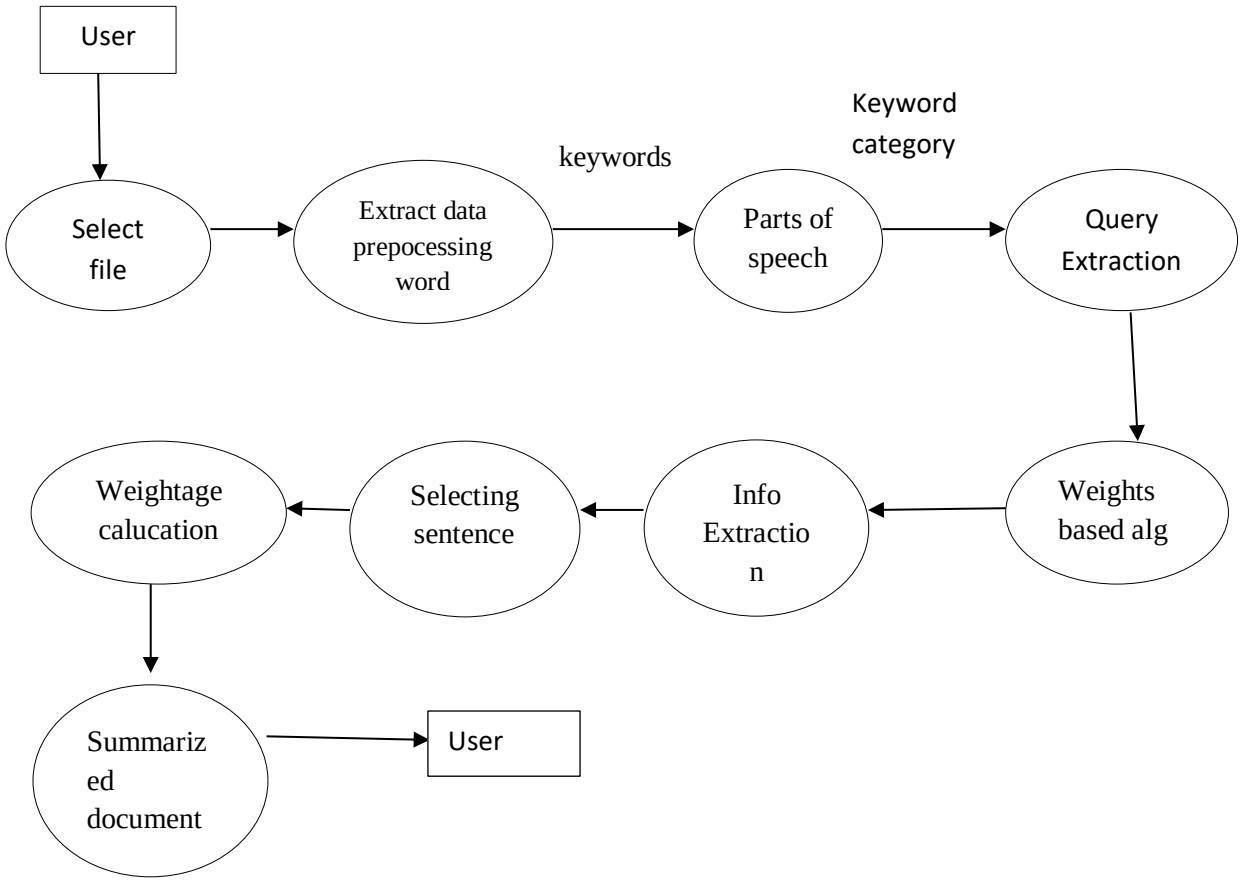


Figure 5.3: Data Process

5.4 CONTROL FLOW

The session of activity that a user with a unique IP address spends on a Web site during a specified period of time. The number of user sessions on a site is used in measuring the amount of traffic a Web site gets. The site administrator determines what the time frame of a user session will b. If the visitor comes back to the site within that time period, it is still considered one user session because any number of visits within that 30 minutes will only count as one session.

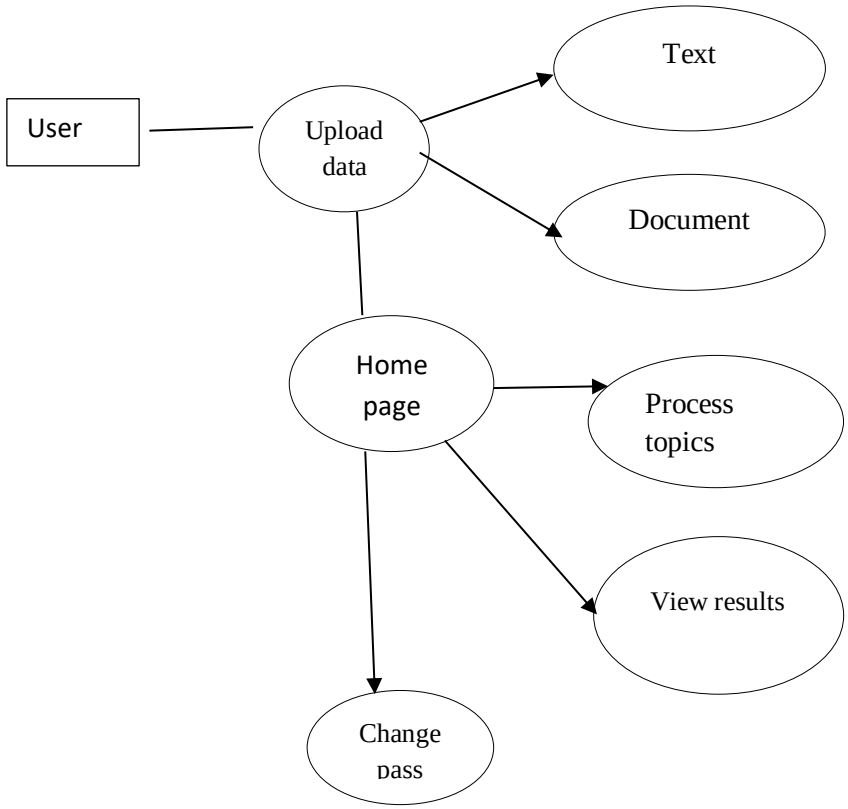


Figure 5.4: DFD of User Session

5.5 USE CASE DIAGRAM

A use case diagram is a dynamic or behavior diagram in UML. Use case diagrams model the functionality of a system using actors and use cases. Use cases are a set of actions, services, and functions that the system needs to perform.

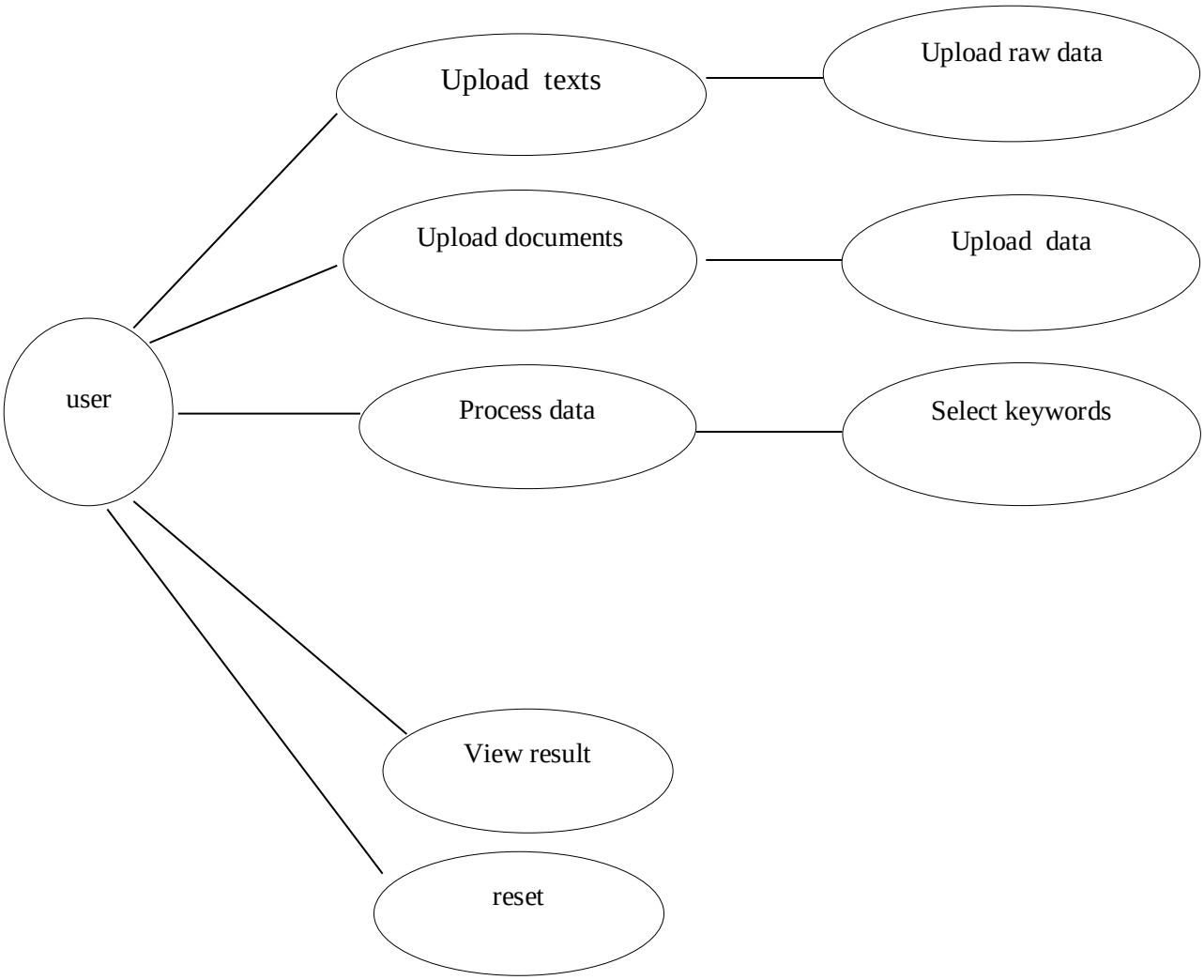


Figure 5.5: Use case diagram for user

The purpose of the use case diagram is as follows:

- Used to gather requirements of a system.
- Used to get an outer view of the system.
- Identify the external and internal view of the system.
- Exhibit the interactions among the requirements and factors.

5.6 MODEL ARCHITECTURE

5.6.1 DATA PRE-PROCESSING

Programmed record outline generator is for clearing the undesirable things which exist in the substance. Henceforth it will additionally process it will be performing sentence part, tokenisation, empty stop word, clear accentuation and perform stemming.

5.6.2 EVALUATION OF WEIGHTS

This stage processes the repeat of the sentences of a substance utilizing lesk count and wordnet. In the first place finding the total number of spreads between a particular and the radiance this philosophy is performed for the all n number of sentences. By then once-over a particular sentence of the substance is set up for each of the sentences. A sentence is snatched from the once-over. Stop words are removing from the sentence as they don't take an intrigue particularly in sense task method. Sparkles of each vital word removed using wordnet. Union is performed between the sparkles and the data content itself. Once-over of all the intersection guide comes to fruition talks toward the largeness of the sentence.

5.6.3 SUMMARIZATION

This stage evaluates the last outline of a substance and the introductions the yield, which is surveyed at the period of arranging the sentences. In the first place it selects the once-over of weight named sentences are planned in jumping demand concerning their weights. Pined for number of sentences is picked by the rate of summary. Picked sentences are re-composed by their genuine gathering in the information content. The modified substance summary will gather a substance without depending

upon the association of the substance, rather than the semantic information lying in the sentence. Modified substance once-over is without vernacular. To remove the semantic information from a sentence, only a semantic word reference in the last vernacular is required.

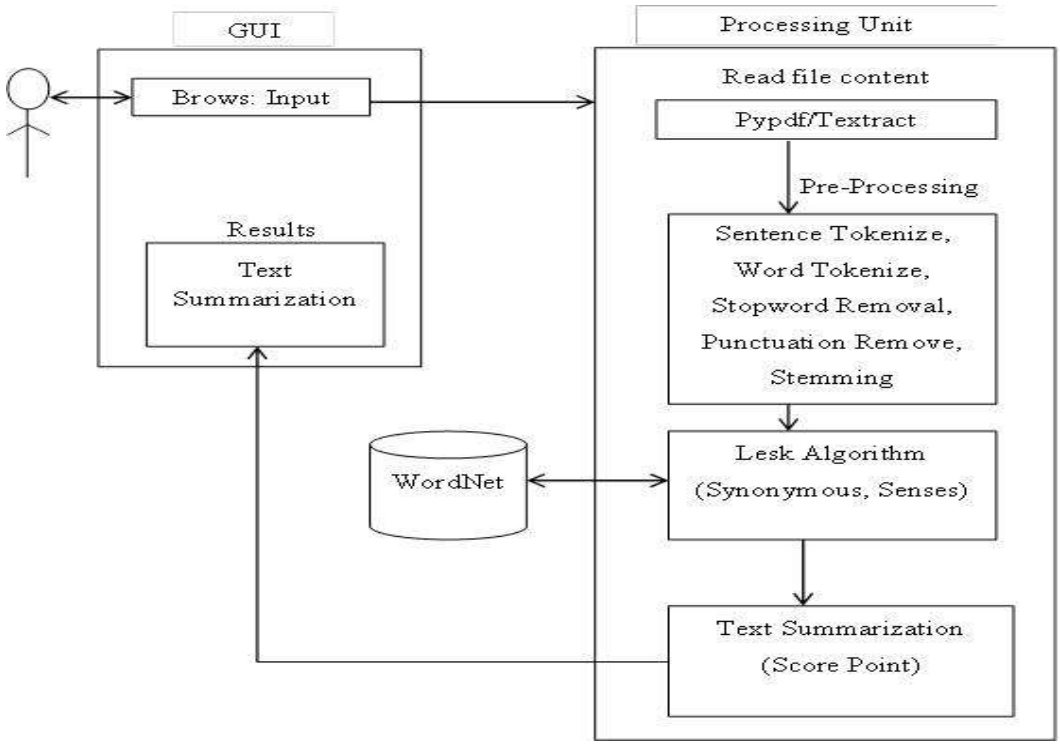


Figure 5.6: System Architecture for Automatic Text Summarization Using Common Handling Dialect

The summarizer calculates score of each feature for every sentence. It uses Apache OpenNLP for name finder(dataset - en-ner-person.bin), location finder(dataset - en-ner- location.bin), and data/time finder(dataset - en-ner-person.bin, data/en-ner-time.bin). After getting scores, it adds them up by giving equal weightage to each feature value using equa- tion 3.9 and returns final score and based upon which the summary is generated

f1 = **Sentence Position**: Sentences appearing in the beginning and at the end of the document are given higher weightage.

f2 = **Positive keyword in the sentence**: It is the frequency of keyword in the summary.

f3 = **Negative keyword in the sentence**: The keywords that are unlikely to appear in summary.

f4 = **Sentence Centrality**: It is the overlap in vocabulary of the sentence and rest of the document. It demonstrates similarity of the sentence with the document.

f5 =**Sentence Resemblance to the title**: It is the overlap in vocabulary between the sentence and the title of the document.

5.7 ALGORITHMS USED IN MODEL

5.7.1 AUTOMATIC SUMMARIZATION BASED ON USER QUERY

Text summarization systems usually provide the user with a generic summary that highlights the most salient information in a text. A Question-Answering (QA) system, however, tries to find an exact answer to the user's query and generate a suitable response to the query. In our system we are using three different techniques to generate a query-based summarization and present the best summary, according to a modified version of the evaluation measure to the user.

We are using a query modification technique to add extra information to the query. The algorithm of this approach is as follows:

1. Generate a Document Graph (DG) for each sentence in the input documents,
2. Generate a DG for the query (topic),
3. Measure the similarity between each sentence and the query (topic),
4. Search for and add the best sentence to the summary,
5. If the summary's length restriction is met or there are no more sentences to add then finish and report the target summary; otherwise add the DG for the chosen sentence to the query graph,
6. Repeat from step 3 until no more sentences can be added to the summary.

In this approach, every time we add a sentence to the target summary, we extend the query graph by adding the sentence's DG to it. We called this new summarizer the Q Inc-summarizer. The "Inc" stands for increment, since in this approach we increment the query graph every time we add a sentence to the summary.

5.7.2 AUTOMATIC SUMMARIZATION BASED ON WEIGHTING

This calculation compresses multiple report content utilizing unsupervised learning approach. In This approach, the heaviness of each sentence in a content is determined utilizing Improved Lesk calculation and WordNet. The summarization procedure is performed as indicated by the given level of summarization

Info: Multiple-report input content.

Yield: Summarized content.

Step 1: The list of distinct sentences of the content is prepared.

Step 2: Repeat steps 3 to 7 for each of the sentences.

Step 3: A sentence is gotten from the list.

Step 4: Stop words are expelled from the sentence as they don't take an interest straightforwardly in sense assessment system.

Step 5: Glosses (dictionary definitions) of all the important words are extricated utilizing the WordNet.

Step 6: Intersection is performed between the sparkles and the information content itself.

Step 7: Summation of all the crossing point comes about speaks to the heaviness of the sentence.

Step 8: Weight appointed sentences are arranged in descending request concerning their weights.

Step 9: Desired number of sentences are chosen by the level of summarization.

Step 10: Selected sentences are re-orchestrated by their real sequence in the info content.

Step 11: Stop.

5.7.3 INFORMATION EXTRACTION ALGORITHM

- Introduce a method to extract the merited keyphrases from the source document. For example, you can use part-of-speech tagging, words sequences, or other linguistic patterns to identify the keyphrases.
- Gather text documents with positively labeled keyphrases. The keyphrases should be compatible to the stipulated extraction technique. To increase accuracy, you can also create negatively-labeled keyphrases.
- Train a binary machine learning classifier to make the text summarization. Some of the features you can use include:
    - Length of the keyphrase
    - Frequency of the keyphrase
    - The most recurring word in the keyphrase
    - Number of characters in the keyphrase
- Finally, in the test phrase, create all the keyphrase words and sentences and carry out classification for them.

CHAPTER 6

IMPLEMENTATION

Implementation is the realization of an application, or execution of a plan, idea, model, design, specification, standard, algorithm, or policy. Realization of a technical specification or algorithm as a program, software component,or other computer system through computer programming and deploymnt.

6.1 HTML

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML, but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference.

6.1.1 BASIC HTML TAGS:

<HTML>….<HTML>	Container tag for html
<! -- -->	specifies comments
<A>…….</A>	Creates hypertext links
<B>…….</B>	Formats text as bold
<BIG>…….</BIG>	Formats text in large font.
<BODY>…</BODY>	Contains all tags and text in the HTML document
<CENTER>...</CENTER>	Creates text

6.1.2 ATTRIBUTES

The attributes of an element are name-value pairs, separated by "=",and written within the

start label of an element, after the element's name. The value should be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML).Leaving attribute values unquoted is considered unsafe. Most elements take any of several common attributes: id, class, style and title. Most also take language-related attributes: lang and dir.

6.2 PYTHON

Python is a multi-paradigm programming language. It supports object-oriented programming, structured programming, and functional programming patterns. It supports object-oriented programming, structured programming, and functional programming patterns. Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a semicolon at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

6.3 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.

The **Natural Language Toolkit**, or more commonly **NLTK**, is a suite of libraries and programs for symbolic and statistical natural language processing (NLP) for English written in the Python programming language. It was developed by Steven Bird and Edward Loper in the Department of Computer and Information Science at the University of Pennsylvania. NLTK includes graphical demonstrations and sample data. It is accompanied by a book that explains the underlying concepts behind the language processing tasks supported by the toolkit, plus a cookbook.

NLTK is intended to support research and teaching in NLP or closely related areas, including empirical linguistics, cognitive science, artificial intelligence, information retrieval, and machine

learning. NLTK has been used successfully as a teaching tool, as an individual study tool, and as a platform for prototyping and building research systems. There are 32 universities in the US and 25 countries using NLTK in their courses. NLTK supports classification, tokenization, stemming, tagging, parsing, and semantic reasoning functionalities.

6.4 WORDNET

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called synsets, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. its primary use is in automatic text analysis and artificial intelligence applications. The most recent Windows version of WordNet is 2.1, released in March 2005. Version 3.0 for Unix/Linux/Solaris/etc. was released in December 2006. Version 3.1 is currently available only online.

WordNet is a lexical database for the English language. It groups English words into sets of synonyms called *synsets*, provides short definitions and usage examples, and records a number of relations among these synonym sets or their members. WordNet can thus be seen as a combination of dictionary and thesaurus. While it is accessible to human users via a web browser, its primary use is in automatic text analysis and artificial intelligence applications. The database and software tools have been released under a BSD style license and are freely available for download from the WordNet website. Both the lexicographic data (*lexicographer files*) and the compiler (called *grind*) for producing the distributed database are available.

WordNet was created in the Cognitive Science Laboratory of Princeton University under the direction of psychology professor George Armitage Miller starting in 1985 and has been directed in recent years by Christiane Fellbaum. The project was initially funded by the U.S. Office of Naval Research and later also by other U.S. government agencies including the DARPA, the National Science Foundation, the Disruptive Technology Office (formerly the Advanced Research and Development Activity), and REFLEX. George Miller and Christiane Fellbaum were awarded the 2006 Antonio Zampolli Prize for their work with WordNet.

Both nouns and verbs are organized into hierarchies, defined by hypernym or *IS A* relationships. For instance, one sense of the word *dog* is found following hypernym hierarchy; the words at the same level represent synset members. Each set of synonyms has a unique index.

At the top level, these hierarchies are organized into 25 beginner "trees" for nouns and 15 for verbs (called *lexicographic files* at a maintenance level). All are linked to a unique beginner synset, "entity". Noun hierarchies are far deeper than verb hierarchies

Adjectives are not organized into hierarchical trees. Instead, two "central" antonyms such as "hot" and "cold" form binary poles, while 'satellite' synonyms such as "steaming" and "chilly" connect to their respective poles via a "similarity" relations. The adjectives can be visualized in this way as "dumbbells" rather than as "trees".

WordNet is sometimes called an ontology, a persistent claim that its creators do not make. The hypernym/hyponym relationships among the noun synsets can be interpreted as specialization relations among conceptual categories. In other words, WordNet can be interpreted and used as a lexical ontology in the computer science sense. However, such an ontology should be corrected before being used, because it contains hundreds of basic semantic inconsistencies; for example, there are common specializations for exclusive categories and (ii) redundancies in the specialization hierarchy. Furthermore, transforming WordNet into a lexical ontology usable for knowledge representation should normally also involve (i) distinguishing the specialization relations into *subtypeOf* and *instanceOf* relations, and (ii) associating intuitive unique identifiers to each category. Although such corrections and transformations have been performed and documented as part of the integration of WordNet 1.7 into the cooperatively updatable knowledge base of WebKB-2, most projects claiming to re-use WordNet for knowledge-based applications (typically, knowledge-oriented information retrieval) simply re-use it directly.

WordNet has also been converted to a formal specification, by means of a hybrid bottom-up top-down methodology to automatically extract association relations from WordNet, and interpret these associations in terms of a set of conceptual relations, formally defined in the DOLCE foundational ontology.

In most works that claim to have integrated WordNet into ontologies, the content of WordNet has not simply been corrected when it seemed necessary; instead, WordNet has been heavily re-interpreted and updated whenever suitable. This was the case when, for example, the top-level ontology of WordNet was re-structured according to the Uncleansed approach or when WordNet was used as a primary source for constructing the lower classes of the SENSUS ontology.

6.5 MONTY LINGUA

Montylingua is a free*, commonsense-enriched, end-to-end natural language understander for English. Feed raw English text into montylingua, and the output will be a semantic interpretation of that text. Perfect for information retrieval and extraction, request processing, and question answering. From English sentences, it extracts subject/verb/object tuples, extracts adjectives, noun phrases and verb phrases, and extracts people's names, places, events, dates and times, and other semantic information.

Montylingua differs from other natural language processing tools because:

- It is complete *end-to-end*.. Input raw_text; output semantic interpretation
- Not many dated tools and implementations sewn together; it is one well-integrated implementation
- It does not require "training" and other fidgetting, and will work right out-of-the-box
- It is enriched with "common sense" knowledge about the everyday world, allowing it to escape many stupid interpretive mistakes. *E.g.:*
    - "(NX the/DT mosquito/NN bit/NN NX) (NX the/DT boy/NN NX)" ==corrected==>
    - "(NX the/DT mosquito/NN NX) (VX bit/VBD VX) (NX the/DT boy/NN NX)"
- It is lightweight and portable across platforms, written in portable Python and also available as a compiled Java library
- It is easy to customize by allowing for a user lexicon

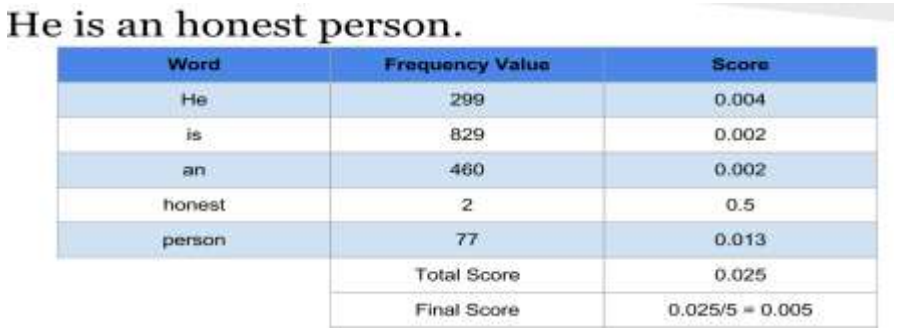
Montylingua performs the following tasks over text:

- Montytokenizer - Tokenizes raw English text (sensitive to abbreviations), and resolve contractions, e.g. "you're" ==> "you are"
- Montytagger - Part-of-speech tagging based on Brill94, enriched with common sense.
- Montychunker - Lightning fast regular expression chunker
- Montyextractor - Extracts phrases and subject/verb/object triplets from sentences
- Montylemmatiser - Strips inflectional morphology, i.e. Changes verbs to infinitive form and nouns to singular form
- Montynlgenerator - Uses montylingua's concise predicate-arg representation to generate naturalistic English sentences and text summaries

6.6 WORKING OF THE MODEL

The summarizer fetches data from the uploaded document and filters out tags, references and other meta content from it. The processed input page is then broken-down section. Each section maintains a separate list of sentences contained in it. After processing and storing of page content, summarizer calculates feature values for each sentence. List of features used are as follow:

- **Sentence Position**: It is a traditional method for providing score for the sentences [2]. Each sentence in a section is given a score based on its relative position in its section. Sentences appearing in the beginning of the section are given higher weightage
- **TF-IDF**: TF-IDF for a word in a sentence is inversely proportional to the number of documents which also contain that word. Words with high TF-IDF numbers imply a strong relationship with the sentence they appear in, suggesting that if that word were to appear in a sentence, it could be interest to the user [5]. Data frequency values are constructed using 1000 randomly selected data. Higher the frequency of the word in corpus, lower will be its value.


He is an honest person.

Word	Frequency Value	Score
He	299	0.004
is	829	0.002
an	460	0.002
honest	2	0.5
person	77	0.013
	Total Score	0.025
	Final Score	0.025/5 = 0.005

Figure 6.6: TF-IDF Illustration

- **Word Similarity**: This feature provides customizability to the summarizer, in which a user can select keywords that he wants/doesn't want in the summary. The summarizer then compares similarity of that word with every word in the sentence using Wordnet dataset. If a sentence contains similar words then depending upon the preference of user it scores the sentence. For positive keywords, it increments the score for sentences containing similarly words on the other hand for negative keywords, it decrements the score for sentences containing similarly keywords

Word1	Word2	Score	Explaination
cat	tiger	0.5	Both belong to *Felidae family*.
apple	tiger	0.1	Less related
apple	orange	0.25	Both are fruits
cat	cat	1.0	Same words
cat	feline	0.5	Synonyms
history	past	0.5	Both referring to time

Table 6.1: Illustration of Word Similarity

- **Text Processor**: After receiving data from the extractor, it removes various content including references, tags, tables, and unwanted sections. Then it breaks down the article into section using regular expressions
- **Summarizer**: After receiving processed data, it starts evaluating scores of sentences for different feature. It concurrently calculates sentences' score for faster processing. It eventually calculates score of each sentences

Chapter 7

TESTING

7.1 INTRODUCTION

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is a process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail in an unacceptable manner.

7.2 FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called feasibility Study. This type of study if a project could be taken. In the conduct of the feasibility study, the analyst will usually consider seven distinct, but inter-related types of feasibility.

7.2.1 TECHNICAL FEASIBILITY

This is considered with specifying equipment and software that will successful satisfy the user requirement the technical needs of the system may vary considerably but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain column of transaction at a particular speed.

7.2.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost / benefit analysis. The procedure is to determine the benefits and savings are expected form a proposed system and a compare them with costs. It benefits outweigh costs; a decision is taken to design and implement the system will have to be made if it is to have a

chance of being approved. There is an ongoing effort that improves in accuracy at each phase of the system life cycle.

7.2.3 OPERATIONAL FEASIBILITY

It is mainly related to human organization and political aspects. These points are considered are :

- What changes will be brought with the system?
- What organizational structures are distributed?
- What new skills will be required?
- Do the existing system staff members have these skills?
- If not, can they be trained in the course of time?

7.3 LEVELS OF TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product.

7.3.1 UNIT TESTING

It deals with testing a unit as a whole. This would test the interaction of many functions but confine the test within one unit. The exact scope of a unit is left to interpretation. Supporting test code, sometimes called Scaffolding, may be necessary to support an individual test. This type of testing is driven by the architecture and implementation teams. This focus is also called black-box testing because only the details of the interface are visible to the test. Limits that are global to a unit are tested here.

In the construction industry, scaffolding is a temporary, easy to assemble and disassemble, frame placed around a building to facilitate the construction of the building. The construction workers first build the scaffolding and then the building. Later the scaffolding is removed, exposing the completed building similarly, in software testing, one particular test may need some supporting software. This software establishes can a correct evaluation of the test take place.

The scaffolding software may establish state and values for data structures as well as providing dummy external functions for the test. Different scaffolding software may be needed form one test to another test. Scaffolding software rarely is considered part of the system. Sometimes the scaffolding software becomes larger than the system software being tested.

Usually the scaffolding software is not of the same quality as the system software and frequently is quite fragile. A small change in test may lead to much larger changes in the scaffolding.

7.3.2 INTEGRATION TESTING

This can proceed in a number of different ways, which can be broadly characterized as top down or bottom up. In top down integration testing the high level control routines are tested first, possibly with the middle level control structures present only as stubs. Subprogram stubs were presented in section2 as incomplete subprograms which are only present to allow the higher. Level control routines to be tested.

- **Top down testing** can proceed in a depth-first or a breadth-first manner. For depth-first integration each module is tested in increasing detail, replacing more and more levels of detail with actual code rather than stubs. Alternatively breadth-first would processed by refining all the modules at the same level of control throughout the application .in practice a combination of the two techniques would be used. At the initial stages all the modules might be only partly functional, possibly being implemented only to deal with non-erroneous data. These would be tested in breadth-first manner, but over a period of time each would be replaced with successive refinements which were closer to the full functionality
- **Bottom Up Testing** where an individual module is tested form a test harness. Once a set of individual module have been tested they are then combined into a collection of modules, known as builds, which are then tested by a second test harness. This process can continue until the build consists of the entire application. The sub teams or individuals would conduct bottom-up testing of the modules which they were constructing before releasing them to an integration team which would assemble them together for top-down testing.

7.3.3 VALIDATION AND SYSTEM TESTING

Validation testing is a concern which overlaps with integration testing. Ensuring that the application fulfils its specification is a major criterion for the construction of an integration test. Validation testing also overlaps to a large extent with System Testing, where the application is tested with respect to its typical working environment. Consequently for many processes no clear division between validation and system testing can be made. Specific tests which can be performed in either or both stages include the following.

- **Regression Testing:** Where this version of the software is tested with the automated test harness used with previous versions to ensure that the required features of the previous version are skill working in the new version.
- **Performance testing:** Where the performance requirements, if any, are checked. These may include the size of the software when installed, type amount of main memory and/or secondary storage it requires and the demands made of the operating when running with normal limits or the response time.
- **Usability Testing:** The process of usability measurement was introduced in the previous chapter. Even if usability prototypes have been tested whilst the application was constructed, a validation test of the finished product will always be required.

Input: Article 1st sentence	Model-written headline
metro-goldwyn-mayer reported a third-quarter net loss of dlrs 16 million due mainly to the effect of accounting rules adopted this year	mgm reports 16 million net loss on higher revenue
starting from july 1, the island province of hainan in southern china will implement strict market access control on all incoming livestock and animal products to prevent the possible spread of epidemic diseases	hainan to curb spread of diseases
australian wine exports hit a record 52.1 million liters worth 260 million dollars (143 million us) in september, the government statistics office reported on monday	australian wine exports hit record high in september

Figure 5: Model testing on other articles

Chapter 8

RESULT

8.1 SUMMARIZATION USING QUERY-BASED ALGORITHM INPUT:

Please enter a query: friend

Output:

Processed query: [u'friend']

[("She was ransacking the stores for Jim's present.", 1.324313323278029), ('Jim stepped inside the door, as immovable as a setter at the scent of quail.', 1.3260330166909116), ("'Give it to me quick" said Della.', 1.337516806722689), ('Jim had not yet seen his beautiful present.', 1.3436049382716049), ("'You needn\'t look for it," said Della.', 1.3593439153439153), ('And then Della leaped up like a little singed cat and cried, "Oh, oh!"', 1.3629012660542072), ('Only $1.87 to buy a present for Jim.', 1.4183846153846156), ('It surely had been made for Jim and no one else.', 1.444810744810745), ("'Jim, darling," she cried, "don\'t look at me that way.', 1.5813651999874911), ("One was Jim's gold watch that had been his father's and his grandfather's.", 1.6168950552674828)]

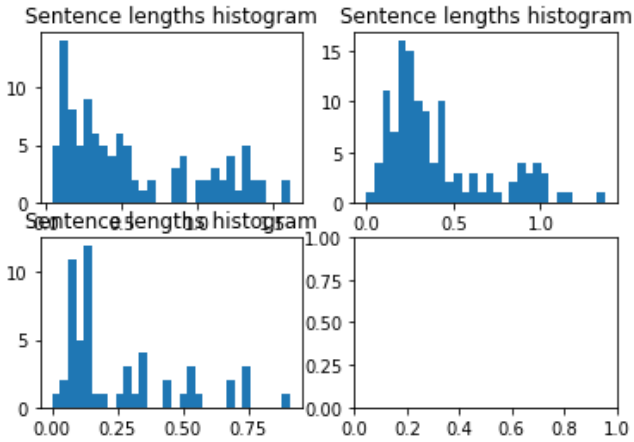


Figure 8.1: Sentence length histogram for query-based output

8.2 AUTOMATIC SUMMARIZATION BASED ON WEIGHTING

Output:

[('Jim drew a package from his overcoat pocket and threw it upon the table.', 16.0), ("'You needn\'t look for it," said Della.', 16.0), ('The door opened and Jim stepped in and closed it.', 16.0), ('It surely had been made for Jim and no one else.', 17.0), ('Jim had not yet seen his beautiful present.', 17.0), ('And then Della leaped up like a little singed cat and cried, "Oh, oh!"', 17.0), ('Only $1.87 to buy a present for Jim.', 17.0), ('Jim looked about the room curiously.', 17.0), ("'Jim, darling," she cried, "don\'t look at me that way.', 18.0), ("One was Jim's gold watch that had been his father's and his grandfather's.", 18.0)]

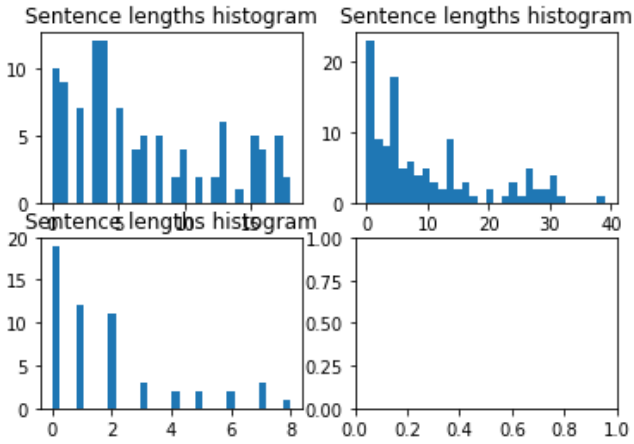


Figure 8.2: Sentence length histogram for Weight-based output

8.3 SUMMARIZATION INFORMATION EXTRACTION

Output: Separation between Subject verb Object

(NX ONE/CD DOLLAR/NN AND/CC EIGHTY-SEVEN/NNP CENTS/NNP NX) ./.(NX THAT/WDT NX) (VX WAS/VBD VX) ALL/PDT ./.(NX AND/CC SIXTY/CD CENTS/NNP NX) of/IN (NX it/PRP NX) (VX was/VBD VX) in/IN (NX pennies/NNS NX) ./.(NX Pennies/NNP NX) (VX saved/VBD VX) (NX one/CD and/CC two/CD NX) at/IN (NX a/DT time/NN NX) by/IN (VX bulldozing/VBG VX) (NX the/DT grocer/NN NX) and/CC (NX the/DT vegetable/NN man/NN NX) and/CC (NX the/DT butcher/NN NX) until/IN (NX one/CD 's/POS cheeks/NNS NX) (VX

burned/VBN VX) with/IN (NX the/DT silent/JJ imputation/NN NX) of/IN (NX parsimony/NN NX) (NX that/WDT such/JJ close/NN NX) (VX dealing/VBG implied/VBN VX) ./.(NX Three/CD times/NNS Della/NNP NX) (VX counted/VBD VX) (NX it/PRP NX) ./.(NX One/CD dollar/NN and/CC eighty-seven/JJ cents/NNS NX) ./.And/CC (NX the/DT next/JJ day/NN NX) (VX would/MD be/VB VX) (NX Christmas/NNP NX)

Output: Summary

THAT" BE. It was in penny saved one and two at time. Bulldozed grocer. One 's cheek burnt with silent imputation of parsimony. That such close" implied. Three-time Dellum counted it. Next day was Christmas.

8.4 SNAPSHOTS

8.4.1 PARTS OF SPEECH TAGGING

```
(NX ONE/CD DOLLAR/NN AND/CC EIGHTY-SEVEN/NNP CENTS/NNP NX) ./.
(NX THAT/WDT NX) (VX WAS/VBD VX) ALL/PDT ./.
(NX AND/CC SIXTY/CD CENTS/NNP NX) of/IN (NX it/PRP NX) (VX was/VBD VX) in/IN (NX pennies/NNS NX) ./.
(NX Pennies/NNP NX) (VX saved/VBD VX) (NX one/CD and/CC two/CD NX) at/IN (NX a/DT time/NN NX) by/IN (VX bulldozing/
VBG VX) (NX the/DT grocer/NN NX) and/CC (NX the/DT vegetable/NN man/NN NX) and/CC (NX the/DT butcher/NN NX) until/IN
(NX one/CD 's/POS cheeks/NNS NX) (VX burned/VBN VX) with/IN (NX the/DT silent/JJ imputation/NN NX) of/IN (NX
parsimony/NN NX) (NX that/WDT such/JJ close/NN NX) (VX dealing/VBG implied/VBN VX) ./.
(NX Three/CD times/NNS Della/NNP NX) (VX counted/VBD VX) (NX it/PRP NX) ./.
(NX One/CD dollar/NN and/CC eighty-seven/JJ cents/NNS NX) ./.
And/CC (NX the/DT next/JJ day/NN NX) (VX would/MD be/VB VX) (NX Christmas/NNP NX) ./.
(NX There/EX NX) (VX was/VBD clearly/RB VX) (NX nothing/NN NX) (VX left/VBN to/TO do/VB VX) (NX but/CC flop/NN NX)
down/RB on/IN (NX the/DT shabby/JJ little/JJ couch/NN and/CC howl/NN NX) ./.
So/RB (NX Della/NNP NX) (VX did/VBD VX) (NX it/PRP NX) ./.
(NX Which/WDT NX) (VX instigates/VBZ VX) (NX the/DT moral/JJ reflection/NN NX) (NX that/DT life/NN NX) (VX is/VBZ
made/VBN VX) up/IN of/IN (NX sobs/NNS NX) ,/, (VX sniffles/VBZ VX) (NX ,/, and/CC smiles/NNS NX) ,/, with/IN (VX
sniffles/VBZ predominating/VBG VX) ./.
While/IN (NX the/DT mistress/NN NX) of/IN (NX the/DT home/NN NX) (VX is/VBZ gradually/RB subsiding/VBG VX) from/IN
(NX the/DT first/JJ stage/NN NX) to/TO the/DT (NX second/JJ NX) ,/, (VX take/VB VX) (NX a/DT look/NN NX) at/IN (NX
the/DT home/NN NX) ./.
A/DT (VX furnished/VBN VX) (NX flat/JJ NX) at/IN (NX $8/CD NX) per/IN (NX week/NN NX) ./.
(NX It/PRP NX) (VX did/VBD not/RB exactly/RB beggar/VB VX) (NX description/NN NX) ,/, but/CC (NX it/PRP NX) (VX
certainly/RB had/VBD VX) (NX that/DT word/NN NX) on/IN (NX the/DT look-out/NN NX) for/IN (NX the/DT mendicancy/NN
squad/NN NX) ./.
In/IN (NX the/DT vestibule/NN NX) below/IN (VX was/VBD VX) (NX a/DT letter-box/NN NX) into/IN (NX which/WDT NX) (NX
no/DT letter/NN NX) (VX would/MD go/VB VX) ,/, and/CC (NX an/DT electric/JJ button/NN NX) from/IN (NX which/WDT NX)
(NX no/DT mortal/JJ finger/NN NX) (VX could/MD coax/VB VX) (NX a/DT ring/NN NX) ./.
(VX Also/RB appertaining/VBG VX) (NX thereunto/NN NX) (VX was/VBD VX) (NX a/DT card/NN NX) (VX bearing/VBG VX) (NX
the/DT name/NN NX) "/" (NX Mr./NNP NX)
```

Figure 8.3: Parts Of Speech Tagging

8.4.2 ASSIGNING THE WEIGHTS TO THE SENTENCE

{'A furnished flat at $8 per week.': 5.0, 'And then she did it up again nervously and quickly.': 1.0, 'Which is all very good.': 1.0, '"You\'ve cut off your hair?"': 4.0, 'Jim stepped inside the door, as immovable as a setter at the scent of quail.': 16.0, 'he said, with an air almost of idiocy.': 2.0, 'It was like him.': 2.0, 'Her Jim.': 13.0, 'Della, being slender, had mastered the art.': 11.0, 'I sold the watch to get the money to buy your combs.': 9.0, 'Rapidly she pulled down her hair and let it fall to its full length.': 7.0, 'It surely had been made for Jim and no one else.': 17.0, 'A mathematician or a wit would give you the wrong answer.': 3.0, '"Dell," said he, "let\'s put our Christmas presents away and keep \'em a while.': 10.0, 'Della wriggled off the table and went for him.': 13.0, "It's Christmas Eve, boy.": 3.0, '"Cut it off and sold it," said Della.': 15.0, '"Isn\'t it a dandy, Jim?': 13.0, "The other was Della's hair.": 13.0, 'Hair Goods of All Kinds."': 4.0, 'It reached below her knee and made itself almost a garment for her.': 3.0, 'Which is always a tremendous task dear friends--a mammoth task.': 2.0, 'Jim drew a package from his overcoat pocket and threw it upon the table.': 16.0, '"Give it to me quick" said Della.': 15.0, 'And now suppose you put the chops on."': 2.0, "Jim, and let's be happy.": 15.0, 'They are the magi.': 1.05, 'AND SIXTY CENTS of it was in pennies.': 3.0, 'Jim had not yet seen his beautiful present.': 17.0, 'She had been saving every penny she could for months, with this result.': 3.0, 'Oh, and the next two hours tripped by on rosy wings.': 2.0, 'She looked at her reflection in the mirror long, carefully, and critically.': 6.0, "As soon as she saw it she knew that it must be Jim's.": 13.0, 'Suddenly she whirled from the window and stood before the glass.': 3.0, 'Be good to me, for it went for you.': 3.0, 'Expenses had been greater than she had calculated.': 1.0, 'And the next day would be Christmas.': 5.0, '"Jim, darling," she cried, "don\'t look at me that way.': 18.0, 'One Eight up Della ran, and collected herself, panting.': 12.0, 'White fingers and nimble tore at the string and paper.': 0.0, "She was ransacking the stores for Jim's present.": 15.0, 'Many a happy hour she had spent planning for something nice for him.': 3.0, 'Everywhere they are wisest.': 1.0, 'I\'m me without my hair, ain\'t I?"': 4.0, 'And then Della leaped up like a little singed cat and cried, "Oh, oh!"': 17.0, '"Twenty dollars," said Madame, lifting the mass with a practised hand.': 7.0, '"Don\'t make any mistake, Dell," he said, "about me.': 4.0, '"You needn\'t look for it," said Della.': 16.0, 'Forget the hashed metaphor.': 0.0, 'The magi brought valuable gifts, but that was not among them.': 4.1, 'The door opened and Jim stepped in and closed it.': 16.0, 'THAT WAS ALL.': 0.0, 'Jim was never late.': 13.0, 'Down rippled the brown cascade.': 4.0, 'My hair grows awfully fast.': 5.0, "You'll have to look at the time a hundred times a day now.": 7.0, 'asked Della.': 10.0, 'Out of his trance Jim seemed quickly to wake.': 14.0, '"You say your hair is gone?"': 4.0, 'Of all who give and receive gifts, such as they are wisest.': 6.05, '"Take yer hat off and let\'s have a sight at the looks of it."': 6.0, 'He needed a new overcoat and he was with out gloves.': 1.0, 'Where she stopped the sign read: "Mme Sofronie.': 1.0, 'There was a pier-glass between the windows of the room.': 3.0, 'ONE DOLLAR AND EIGHTY-SEVEN CENTS.': 8.0, 'One dollar and eighty-seven cents.': 8.0, 'Eight dollars a week or a million a year--what is the difference?': 4.0, 'On went her old brown jacket; on went her old brown hat.': 9.0, "One was Jim's gold watch that had been his father's and his grandfather's.": 18.0, 'Shall I put the chops on, Jim?"': 15.0, '"I buy hair," said Madame.': 8.0, 'I his dark assertion will be illuminated later on.': 0.0, 'He enfolded his Della.': 10.0, "Say 'Merry Christmas!'": 4.0, 'He looked thin and very serious.': 5.0, 'Perhaps you have seen a pier-glass in an $8 flat.': 3.0, 'She held it out to him eagerly upon her open palm.': 0.0, 'what could I do with a dollar and eighty-seven cents?"': 7.0, 'They always are.': 0.0, "It'll grow out again--you won't mind, will you?": 1.0, 'It was even worthy of The Watch.': 4.0, 'I just had to do it.': 0.0, 'I hunted all over town to find it.': 0.0, 'Three times Della counted it.': 11.0, "They're too nice to use just at present.": 5.0, 'And then an ecstatic scream of joy; and then, alas!': 0.0, 'Give me your watch.': 5.0, 'You don\'t know what a nice-what a beautiful, nice gift I\'ve got for you."': 8.05, 'Madame, large, too white, chilly, hardly looked the "Sofronie."': 6.0, 'But what could I do--oh!': 2.0, 'I want to see how it looks on it."': 4.0, 'Quietness and value--the description applied to both.': 1.0, 'So Della did it.': 10.0, '"Don\'t you like me just as well, anyhow?': 2.0, 'They invented the art of giving Christmas presents.': 8.0, "Twenty dollars a week doesn't go far.": 7.0, 'Only $1.87 to buy a present for Jim.': 17.0, '"It\'s sold, I tell you--sold and gone, too.': 4.0, 'He simply stared at her fixedly with that peculiar expression on his face.': 3.0, 'Poor fellow, he was only twenty-two--and to be burdened with a family!': 3.0, 'Della finished her cry and attended to her cheeks with the powder rag.': 12.0, 'Jim looked about the room curiously.': 17.0, '"Will you buy my hair?"': 5.0, 'She found it at last.': 8.0}
[('Jim drew a package from his overcoat pocket and threw it upon the table.', 16.0), ('"You needn\'t look for it," said Della.', 16.0), ('The door opened and Jim stepped in and closed it.', 16.0), ('It surely had been made for Jim and no one else.', 17.0), ('Jim had not yet seen his beautiful present.', 17.0), ('And then Della leaped up like a little singed cat and cried, "Oh, oh!"', 17.0), ('Only $1.87 to buy a present for Jim.', 17.0), ('Jim looked about the room curiously.', 17.0), ('"Jim, darling," she cried, "don\'t look at me that way.', 18.0), ("One was Jim's gold watch that had been his father's and his grandfather's.", 18.0)]

Figure 8.4: Weighted Sentences

8.4.3 COMBINING THE SENTENCES USING HISTOGRAM

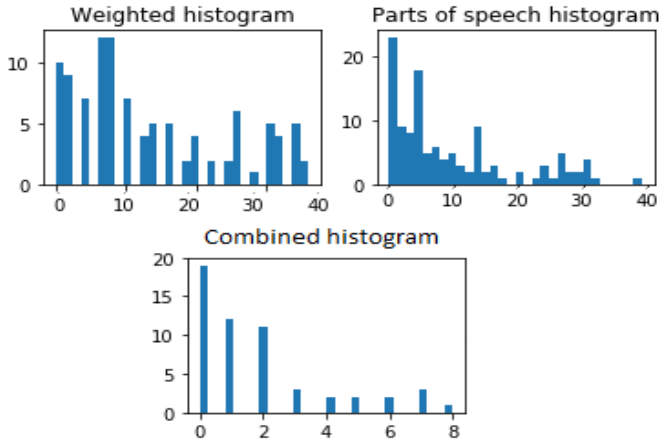


Figure 8.5: Combined Histogram Graph

8.4.4 SUMMERY OF THE FIRST TEST FILE

```
gift of magi
Jim stepped inside the door, as immovable as a setter at the scent of quail.
Her Jim.
Della, being slender, had mastered the art.
I sold the watch to get the money to buy your combs.
It surely had been made for Jim and no one else.
"Dell," said he, "let's put our Christmas presents away and keep 'em a while.
Della wriggled off the table and went for him.
"Cut it off and sold it," said Della.
"Isn't it a dandy, Jim?
The other was Della's hair.
Jim drew a package from his overcoat pocket and threw it upon the table.
"Give it to me quick" said Della.
Jim, and let's be happy.
Jim had not yet seen his beautiful present.
As soon as she saw it she knew that it must be Jim's.
"Jim, darling," she cried, "don't look at me that way.
One Eight up Della ran, and collected herself, panting.
She was ransacking the stores for Jim's present.
And then Della leaped up like a little singed cat and cried, "Oh, oh!"
"You needn't look for it," said Della.
The door opened and Jim stepped in and closed it.
Jim was never late.
asked Della.
Out of his trance Jim seemed quickly to wake.
On went her old brown jacket; on went her old brown hat.
One was Jim's gold watch that had been his father's and his grandfather's.
Shall I put the chops on, Jim?"
He enfolded his Della.
Three times Della counted it.
So Della did it.
Only $1.87 to buy a present for Jim.
Della finished her cry and attended to her cheeks with the powder rag.
Jim looked about the room curiously.
```

Figure 8.6: Summery Of The File Gift Of Magi

8.4.4 SUMMERY OF THE SECOND TEST FILE

```
the skylight room
Miss Leeson sat on the middle step and the men would quickly group around her.
"Excuse me, Mr. Skidder," said Mrs. Parker, with her demon's smile at his pale looks.
sighed Miss Leeson, sinking down upon the squeaky iron bed.
Miss Longnecker must be right; it was Gamma, of the constellation Cassiopeia, and not Billy Jackson.
"He doesn't show up very well from down here," said Miss Leeson.
There came a time after that when Miss Leeson brought no formidable papers home to copy.
said Miss Longnecker.
Every day Miss Leeson went out to work.
"They're too lovely for anything," said Miss Leeson, smiling in exactly the way the angels do.
But Miss Longnecker, the tall blonde who taught in a public school and said, "Well, really!"
It's a young woman, a Miss Elsie--yes, a Miss Elsie Leeson.
said Miss Leeson.
"Same here," said Mr. Hoover, loudly breathing defiance to Miss Longnecker.
One day Miss Leeson came hunting for a room.
Miss Leeson was not intended for a sky-light room when the plans were drawn for her creation.
"It's that star," explained Miss Leeson, pointing with a tiny finger.
"I think Miss Leeson has just as much right to name stars as any of those old astrologers had."
sounded to the world the state of Miss Leeson's purse.
First Mrs. Parker would show you the double parlours.
"I didn't know you were an astronomer, Miss Leeson."
There was rejoicing among the gentlemen roomers whenever Miss Leeson had time to sit on the steps for an hour or two.
If you survived Mrs. Parker's scorn, you were taken to look at Mr. Skidder's large hall room on the third floor.
```

Figure 8.7: Summery Of The File The Skylight Room

8.4.4 SUMMERY OF THE THIRD TEST FILE

```
the cactus
Humbled now, he sought the answer amid the ruins of his self-conceit.
That is what Trysdale was doing, standing by a table in his bachelor apartments.
White favors like stars upon their coats shone through the gloom of the apartment.
The natives imagine the leaves are reaching out and beckoning to you.
Here's the name on this tag tied to it.
Vanity and conceit?
Come now!
Once that same look had been raised to him, and he had gauged its meaning.
"I don't drink just now, thanks," said Trysdale.
He allowed the imputation to pass without denial.
"Your brandy," resumed the other, coming over and joining him, "is abominable.
With womanly swiftness she took her cue from his manner, and turned to snow and ice.
Thus, and wider from this on, they had drifted apart.
On the table stood a singular-looking green plant in a red earthen jar.
Name means in English, 'Come and take me.'"
He waited until night, but her answer did not come.
He saw all the garbs of pretence and egoism that he had worn now turn to rags of folly.
At noon her groom came to the door and left the strange cactus in the red earthen jar.
"No," said Trysdale, with the bitter wraith of a smile--"Is it Spanish?"
Wherever did you rake up this cactus, Trysdale?"
"A present," said Trysdale, "from a friend.
Indeed, his conceit had crumbled; its last prop was gone.
"I say, Trysdale, what the deuce is the matter with you?
Know any Spanish, Trysdale?"
```

Figure 8.6: Summery Of The File The Cactus

CHAPTER 9

CONCLUSIONS AND FUTURE SCOPE

9.1 CONCLUSIONS

Document summarization is growing as sub branch of NLP as the demand for compressive, meaningful, abstract of topic due to large amount of information available on net. Precise information helps to search more effectively and efficiently. Thus Document summarization is need and used by business analyst, marketing executive, development, researchers, government organizations, students and teachers also. It is seen that executive requires summarization so that in a limited time required information can be processed. The summarizer built using query-based algorithm has shown satisfactory results for wide variety of documents. The weight-based algorithm gives the summery according to the score of the sentence but it is not accurate in the formation of the sentences. The information extraction algorithm uses parts of speech tagging which gives the summary having proper meaning for the sentences. Hence, we like to conclude that information extraction algorithm is more accurate than query and weight-based algorithms.

9.2 FUTURE SCOPE

- Building a single system which can automatically identify the type of input text and use the best method out of these three techniques to generate summary.
- Improving accuracy of features like name entity identifier, location finder etc.
- Extending the domains for extraction-based summarizer by training it using various other datasets in diverse domains.
- Adding other languages support for summarization

BIBILIOGRAPHY

[1] G. Sizov, "Extraction-based automatic summarization: Theoretical and empirical in- vestigation of summarization techniques," 2010.

[2] M. A. Fattah and F. Ren, "Automatic text summarization," *World Academy of Science, Engineering and Technology*, vol. 37, p. 2008, 2008.

[3] "About wordnet - wordnet about wordnet," https://wordnet.princeton.edu/, accessed: 2015-05-15.

[4] "word2vec - tool for computing continuous distributed representations of words. - google project hosting," https://code.google.com/p/word2vec/, accessed: 2015-05-15.

[5] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proceed- ings of the first instructional conference on machine learning*, 2003.

[6] W. T. Chuang and J. Yang, "Extracting sentence segments for text summarization: a machine learning approach," in *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2000, pp. 152–159.

[7] S. Ryang and T. Abekawa, "Framework of automatic text summarization using rein- forcement learning," in *Proceedings of the 2012 Joint Conference on Empirical Meth- ods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 2012, pp. 256–265.

[8] D. Hingu, D. Shah, and S. S. Udmale, "Automatic text summarization of wikipedia articles," in *Communication, Information & Computing Technology (ICCICT), 2015 International Conference on*. IEEE, 2015, pp. 1–4.

[9] H. P. Edmundson, "New methods in automatic extracting," *Journal of the ACM (JACM)*, vol. 16, no. 2, pp. 264–285, 1969.

[10] Alexander M. Rush Sumit Chopra Jason Weston A Neural Attention Model for Sentence Summarization, Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pages 379–389, Lisbon, Portugal, 17-21 September 2015. c 2015 Association for Computational Linguistics

[11] Deepali K. Gaikwad[1] and C. Namrata Mahender A Review Paper on Text Summarization, International Journal of Advanced Research in Computer and Communication Engineering

[12] elima Bhatia, Arunima jaiswal – Automatic text summarization and its methods- A review , 978-1-4673-8203/16/$31.00 IEEE 2016

[13] Anusha Bagalkotkar, Ashesh Khandelwal, Shivam Pandey, Sowmya Kamath S, A Novel Technique for Efficient Text Document Summarization as a Service 2013 Third International Conference on Advances in Computing and Communications, 978-0-7695-5033-6/13 $26.00 © 2013 IEEE,

[14] Pratibha Devihosur1, Naseer R - Automatic Text Summarization Using Natural Language Processing, International Research Journal of Engineering and Technology (IRJET), e-ISSN: 2395-0056,

[15] Yogan Jaya Kumar, Ong Sing Goh, Halizah Basiron, Ngo Hea Choon and Puspalata C Suppiah- A Review on Automatic Text Summarization Approaches, 2016 Yogan Jaya Kumar, Ong Sing Goh, Halizah Basiron, Ngo Hea Choon and Puspalata C Suppiah. This open access article is distributed under a Creative Commons Attribution (CC-BY) 3.0 license.

[16] Nenkova, Ani, and Kathleen McKeown. Automatic summarization. Now Publishers Inc, 2011.

[17] Mani, Inderjeet, and Mark T. Maybury. Advances in automatic text summarization. the MIT Press, 1999.

[18] Goldstein, Jade, Vibhu Mittal, Jaime Carbonell, and Mark Kantrowitz. "Multi-document summarization by sentence extraction." In Proceedings of the 2000 NAACL-ANLPWorkshop on Automatic summarization-Volume 4, pp. 40-48. Lal, Partha. "Text Summarization." (2002).

[19] Salton, Gerard, and Christopher Buckley. "Term-weighting approaches in automatic text retrieval." Information processing & management (1988)

[20] Kumar Nagwani, Naresh, and Shrish Verma. "A Frequent Term and Semantic Similarity based Single Document Text Summarization Algorithm." International Journal of Computer Applications (2011)

[21] Yang, Guangbing, Wen, Nian-Shing, and Sutinen. "Personalized Text Content Summarizer for Mobile Learning: An Automatic Text Summarization System with Relevance Based Language

Model." InTechnology for Education (T4E), 2012 IEEE Fourth International Conference on, pp. 90-97. IEEE, 2012.

[22] Aksoy, Bugdayci, Gur, Uysal, and Can. "Semantic argument frequency-based multi-document summarization." In Computer and Information Sciences, 2009. ISCIS 2009. 24th International Symposium on, pp. 460-464. IEEE, 2009.

[23] Shams, Rushdi, M. M. A. Hashem, Suraiya Rumana Akter, and Monika Gope. "Corpus-based web document summarization using statistical and linguistic approach." In Computer and Communication Engineering (ICCCE), 2010 International Conference on, IEEE, 2010.

[24] Foong, Oi-Mean, and Alan Oxley. "A hybrid PSO model in Extractive Text Summarizer." In Computers & Informatics (ISCI), 2011 IEEE Symposium on, pp. 130-134. IEEE, 2011.

[25] Resnik, Philip. "Semantic similarity in a taxonomy: An information-based measure and its application to problems of ambiguity in natural language." arXiv preprint arXiv:1105.5444 (2011)