# ANALYSIS FOR TEXT SUMMARIZATION ALGORITHMS FOR DIFFERENT DATASETS

A Dissertation submitted in partial fulfillment of the requirement for

the award of the degree of

**Master of Technology**

**In**

**Software Engineering**


**Submitted by**

**Swati Singh**

**(Roll No.- 2K15/SWE/20)**

**Under the guidance of**

**Prof. (Dr.) Daya Gupta**




**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**DELHI TECHNOLOGICAL UNIVERSITY**

**(Formerly Delhi College of Engineering)**

**BAWANA ROAD, DELHI**


**2015-2017**

# DECLARATION

I, hereby declare that the work embodied in the dissertation entitled **"ANALYSIS FOR TEXT SUMMARIZATION ALGORITHMS FOR DIFFERENT DATASETS"** towards the partial fulfillment of the requirements for the award of degree of **Master of Technology** with specialization in **Software Engineering** is an authentic record of the work carried out under the supervision of **Dr. Daya Gupta**, Professor, Computer Science and Engineering Department, Delhi Technological University, Delhi.

The matter embodied in this dissertation record has not been submitted by me for the award of any other degree.

SWATI SINGH

2K15/SWE/20

**DELHI TECHNOLOGICAL UNIVERSITY**

*(Formerly Delhi College of Engineering)*

## CERTIFICATE

Date: _____

This is to certify that the work embodied in the thesis entitled **"Analysis For Text Summarization Algorithms For Different Datasets"** submitted by **Swati Singh** with **Roll no. 2K15/SWE/20** in partial fulfillment for the award of the **degree of Master of Technology in SOFTWARE ENGINEERING** is an authentic record of student's own work carried out under my supervision.

This work is original research and has not been submitted, in part or full, to any other University or Institute for the award of any degree.

---

**Supervisor**

**Dr. Daya Gupta**

Professor

Department of Computer Science and Engineering

---

SHAHBAD DAULATPUR, BAWANA ROAD, DELHI-110042, INDIA

OFF.:91-11-27871018 FAX: +91-11-27871023 WEBSITE: www.dtu.ac.in

# ACKNOWLEDGEMENT

I am very thankful to **Dr. Daya Gupta** (Professor, CSE Dept) and all the faculty members of the Dept. of Computer Science and Engineering, DTU. They all provided us with immense support and guidance during the project.

I would also like to express gratitude to Mrs. Divyashikha Sethia(Assistant Professor, Delhi Technological University) for providing me continuous support and guidance during this project.

I would also like to express my gratitude to the University for providing us with the laboratories, infrastructure, testing facilities and environment which allowed us to work without any obstructions.

I would also like to appreciate the support provided by our lab assistants, seniors and our peer group who aided us with all the knowledge they had regarding various topics.

Swati Singh

2K15/SWE/20

# ABSTRACT

With the exponential increase in the data available on the internet for a single domain, it is difficult to understand the gist of a whole document without reading the whole document. Automatic Text Summarization reduces the content of the document by presenting important key points from the data. Extracting the major points from the document is easier and requires less machinery than forming new sentences from the available data. Research in this domain started nearly 50 years ago from identifying key features to rank important sentences in a text document. The main aim of text summarization is to obtain human quality summarization, which is still a distant dream. Abstractive Summarization techniques uses dynamic wordnet corpus to produce coherent and succinct summaries.

Automatic text summarization has applications in various domains including medical research, legal domain, doctoral research, documents available on internet etc. To serve the need of text summarization, numerous algorithms based on different content selection and features using different methodologies are made in last half century. Research started from Single document summarization has shifted to Multi-document summarization in last few decades in order to save more time and compressing the same domain documents at once. Here, An analysis is presented on the Single document and Multi-document summarization algorithms on different domain datasets.

# Table of Contents


**Acknowledgement** **iv**
**Abstract** **v**
**Table of Contents** **vi**
**List of Figures** **vii**
**List of Tables** **viii**

**1. Introduction..... 1**
1.1 Introduction 1
1.2 Motivation 3
1.3 Related Work 3
1.4 Problem statement 4
1.5 Scope of work 5
1.6 Organization of thesis 6
**2. Overview of Automatic Text Summarization..... 7**
2.1 Summarization Features 7
2.2 Summarization Approaches 9
2.3 Methods 9
2.4 Types of Summarization 10
2.5 Summary Techniques 10
2.6 Literature Survey 18
**3. Single Document Summarization Algorithms..... 21**
3.1 TextRank Algorithm 21
3.2 Texteaser 24
3.3 Summary based on Word features 27
**4. Multi Document Summarization Algorithms..... 29**
4.1 Multi document Summarization using ILP based Sentence Compression 29
4.2 Multi document Summarization based on LDA Topic Model 33
4.3 Multi document Summarization based on Sentence Clustering 36
**5. Comparison and Evaluation 44**
5.1 Measures 44
5 .2 Single document Summarization Algorithms 45
5.3 Multi-document Summarization Algorithms 49
**6. Conclusions and Future Scope…..... 52**
**Appendix I……. 54**
**References..... 56**

# List of Figures


Figure 2.1: Taxonomy of Automatic Text Summarization 8

Figure 2.2: Group A heuristics 13

Figure 2.3: Group B heuristics 13

Figure 2.4: Markov model to extract the three summary sentences 14

Figure 3.1: Flow chart of TextRank 23

Figure 3.2: Flowchart of TextTeaser 25

Figure 3.3: Flowchart of Summary based on Word features 28

Figure 4.1 Overview of the Summarization approach 30

Figure 4.2: Flowchart for Multi-document Abstractive Summarization using ILP based Sentence Compression 32

Figure 4.3: Flow-Chart: Multi-Document Summarization based on LDA topic Model 34

Figure 4.4: Steps in Multi Document Algorithms 37

Figure 4.5: Parts of Word Net Hierarchy 41

Figure 4.6: Flowchart of Sentence Clustering Algorithm 43

Figure 5.1: Graphical Representation-News blog- Demonetisation Dataset 46

Figure 5.2: Graphical Representation- Medical- Alzheimer's Dataset 47

Figure 5.3: Graphical Representation- Cricket Related Dataset 48

Figure 5.4: Multi-document Graphical Representation- News blog- Demonetisation Dataset 49

Figure 5.5: Multi-document Graphical Representation- Medical- Alzheimer's Dataset 50

Figure 5.6: Multi-document Graphical Representation- Cricket Related Dataset 51

# List of Tables


| | |
|---|---|
| Table 2.1: A Comparative Study on Text Summarization Methods Based On Method, Features and Content Selection | 19 |
| Table 5.1: Comparison table for News blog- Demonetisation Dataset | 46 |
| Table 5.2: Comparison table for Medical- Alzheimer's Dataset | 47 |
| Table 5.3: Comparison table for Cricket related Dataset | 48 |
| Table 5.4: Multi-document Comparison table for News blog- Demonetisation Dataset | 49 |
| Table 5.5: Multi-document Comparison table for Medical- Alzheimer's Dataset | 50 |
| Table 5.6: Multi-document Comparison table for Cricket related Dataset | 51 |

# Chapter -1

# Introduction

Text summarization aims at shortening a text document. This chapter introduces the work carried out in this thesis. The problem statement is stated along with the reasons that lead to the motivation for working in this field.

## 1.1 Introduction

Text summarization is a method to produce a concise and important piece of information from a larger set of text which can be a text document, an article or a blog. Text Summarization aims to provide a summary of given text while preserving its information and intent. The summary is a small piece of information that describes a set of paragraphs or documents. Summary generated is generally less than forty percent of the original text data and it should be even less than that in the case of large datasets. The summary should retain the important data present in the document, should be controllable, short and succinct. Summarization of text data is done in many ways depending upon the various parameters based on the position and format of words and sentences.

Automatic Text Summarization[1] accumulates the data from several documents to present the final shorter piece of information as a result, which is shorter, informative and preserves the real intent of information. These small summarized versions save valuable time by presenting unambiguous important information. With the increasing amount of digital data, it has become difficult to retrieve the needed and concise information. Automatic text summarization caters to the very need of the time.

There are methods which are helpful to produce a summary. First Division which categorizes the summarization approaches is based on the content of the summary produced. There are two approaches-**Extraction** and **Abstraction[**2**]**. As the name suggests, Extraction is domain independent, it mainly aims at finding out the important sentences and later presenting a set of important sentences as Summary. On the contrary, Abstraction is domain dependent, it processes

the available information and new sentences are prepared by understanding the content, also takes human knowledge into consideration by preparing the goal to produce a summary.

## Text Summarization Applications

Automatic text summarization has many important applications. One of the important application is in Medical area, where a lot of unclassified information is available and many times a medical associate is required to find about some information specific to a medical condition for research or diagnostic purpose from the large heap of documents. Finding out relevant information involves the reading of numerous documents and problem/patient's records. Summarization specifically personalized to the medical area is very useful, as it not only saves time but increases the efficiency of a medical expert.

In legal processes, a typical case study involves consideration of loads of information consisting of law expertise books and numerous related previous judicial case studies, thus leading to an overload of information. The legal experts perform a tedious and responsible task, their time and resources are expensive. To find out an important piece of information unambiguously and in less time is desirable to cater the needs of fast and correct court decisions.

For Research Purposes, hundreds of research papers need to be considered in any research domain to find out a specification. In this way to know what lies inside of the paper, researchers need to read more than the abstract but less than the paper, so summarization may be applied to get the customized summary of the content by applying the desirable method.

On the internet, Summarization is used in multiple applications. Various newspaper sites and related apps provide everyday news with the use of summarization in order to save time and space while keeping the important key information. Mainly editorials are summarized while keeping the intent of author intact.

Further, there are also applications for mobile devices like smartphones, tablets etc. they include small screen area and time available to read. In the corporate world, 'meeting minutes' need to be read in small time and associated documents need to be looked into before next meeting without the help of human and other resources. For blind people, a lot of time can be saved by

readers while reading to them by giving them an important piece of information instead of the whole document.

## 1.2 Motivation

Due to exponential rise of the information on the internet, it is difficult to find the relevant information in the first go. High-Quality search engines find hundreds of suitable pages aligned to the search keywords, but finding the appropriate content in the search results is a very tedious task. Humans can't read hundreds of pages in a day time, so Automatic Text Summarization is the answer here. Automatic Text Summarization has become the need of the hour to filter the information in order to get the relevant piece of data. It enables the reader to go through the essential contents in a short period of time. Huge data available on the internet is required to be compressed so that the user can go through it and never miss the important set of information because of the large size of documents. The research on the text summarization started in the early 1950s at IBM and later Information retrieval systems were designed. The growing volume of online text and the availability of a large number of electronic documents on the internet demand new and more efficient information retrieval systems.

As the research proceeds to the advanced phase and due to the availability of numerous documents on the same genre, the need of multiple document summarization[] has emerged. Multi document summarization generates a concise summary by combining the important data from a set of documents on the similar topic for better analysis and covers a wide perspective on a single topic.

## 1.3 Related Work

A lot of good work has been done on the text Summarization. There are various approaches which are categorized and implemented to perform summarization on text data. In most of the Summarizers, Sentences are considered as a feature vector[] and various algorithms are applied depending upon the position of Sentences, Vocabulary intersections, title distribution and the type of data. Apart from the sentence related data other features include the structure of the document and popularity of the content. As there are two main approaches- extractive and abstractive. In extractive, important sentences are extracted based upon weights assigned. In

Abstractive, new sentences are formed based upon the content of the original document. Most of the work has been done in the extraction domain, but various different ideas have been explored like multiple document summaries, language based summarizers etc.

In 1955, Henry Peter Luhn, IBM inventor first published a paper entitled *'A new method of recording and searching information'* (Luhn, 1953) **[**3**].** He developed many Information retrieval applications. Later in 1969, Edmundson described a new extraction method based on extraction using three components: pragmatic words(cue words), structural indicators and topic heading words[4].

In 1980's AI methods came into existence for summarization using hybrid approaches for different types of summarization i.e. multiple documents, multimedia etc.

One of the applications is KWIC (Keyword in Context) by using three fundamental elements: Keyword, title, and context.

In the last two decades various new and hybrid methods have been described. TextRank[5], cluster based[6], Rhetoric based[7], Topic models[8], ILP based method[11] etc.

In an Abstractive domain, as new sentences need to be designed, it needs a deeper analysis of the original text information. It involves an understanding of the text by linguistic methods[9] to provide an interpretation to match the level of human generated summary. There are two main approaches for doing this i.e. structure based and semantic based approach. In Structure based approach, most weighted data is encoded by cognitive schemas[10]. Structures such as a tree, ontology, lead and body phrase structures are the schemas mostly used for structured approach. In the second approach, semantic based uses a Natural Language generation system to process the semantic information to categorize the grammar variants such as noun and verbs by processing linguistic data. To achieve a true abstractive summarization is still a dream

## 1.4 Problem Statement

We have seen that a lot of work has been done in extractive and abstractive text summarization. In Extractive Summaries, a lot of information is lost because it picks whole sentences without any modifications. Sometimes a sentence containing important information partially is totally ignored. Whereas in Abstractive Summaries, we try to form new, informative and coherent sentences. Here, we have a lot of options to extract summaries. With the availability of numerous

techniques for text summarization, it is difficult to choose any one out of these, which can be used for a particular domain.

Therefore, the problem statement of this thesis is: **"to prepare a comparison of the single document and multiple document algorithms and find out domains on which these algorithms work better than others."** This analysis helps in identification of best suitable algorithm for a particular domain.

## 1.5 Scope of work

Text summarization algorithms shorten the text and include only the vital information. There exist many online text summarizers which implement different algorithms to summarize the given text. This work aims at checking the accuracy of the present day text summarization algorithms. The scope of this work can be summarized as:

- Different algorithms each for single text and multi text summarizers are identified.
- The datasets are based on news domain, medical domain and sports domain.
- The algorithms generated summaries are evaluated against the human generated summaries. Based upon the similarity of the words present in human prepared summary and algorithm generated summary, we will evaluate the ROUGE[12] and Bleu Score[13] to show the relevance of summary produced by the algorithms.

The algorithms used for analysis of single document summarizations are TextRank[5] (which further uses PageRank[14] to select sentences), Texteaser and summary tools based on the word features. The algorithms used for multiple document text summarizations include one based on Latent Dirichlet Allocation (LDA) [8] topic model to find out latent topics and topic distribution to select the sentences for final output summary. Other algorithm for Multi Document summarization which first selects the most important document in a set of documents using LexRank and then forms clusters for each sentence of important document aligning with the sentences of other documents sentences and finally finds a sentence from each cluster using Integer Linear Programming (ILP) method[11] with the aim to maximize the information content and Linguistic Quality Score.

## 1.6 Organization of thesis

This sub heading gives brief details about the chapters in this thesis.

**Chapter 2: Overview of Automatic Text Summarization-** The aim of this chapter is to explain about Automatic Text Summarization in detail. The chapter discusses the available techniques and a brief of work done and then presents a comparative analysis with the help of table discussing the features used for content selection.

**Chapter 3: Single Document Summarization Algorithms–**This chapter includes the description and the flow chart of algorithms which work on Single documents to generate a summary.

**Chapter 4: Multi Document text summarization Algorithms-**In the first part of the chapter, we discuss a Multi-document algorithm based on LDA Topic Model and in the second part, a Multi-document summarization algorithm based on ILP Method and third algorithm based on Sentence Clustering.

**Chapter 5: Comparison and Evaluation-** This chapter defines the measures used to check the accuracy of summaries and comparison tables and graphs for algorithms described in previous chapters.

**Chapter 6: Conclusions and Future Work-** This chapter states the work done in this thesis and future work that can be done on the basis of this work.

# Chapter 2

# Overview of Automatic Text Summarization

This chapter discusses the different features, on the basis of which text documents are summarized.

## 2.1 Features

Automatic Text summarization works on the source documents in order to produce important, non-redundant and useful sentences to form a summary by concatenating them. In order to decide the degree of importance a sentence to include or exclude from the final summary formation, a list of features[] used by the researchers are listed below:

**Term Frequency[19]:** Frequency of a word is measured for the whole source document. Then, the scores are assigned to each sentence based upon the number of frequent words belonging to the particular sentence. Sentences with highest weights are considered for final summary. TF IDF is widely used for calculation of word frequency.

**Location:** In a text document, the position of a sentence also tells about its relevance in the summary. While calculating the weighted Score of a sentence certain sentences are weighted higher than others.

**Cue Method:** Sentences including words that adds to limitations or advantages of the content are weighted high i.e. “in summary”, “significantly”, “describes” “concludes” are cue words.

**Title/ Headline[20]:** words included in the topic or theme of the content are considered relevant. Sentences which includes these words are assumed to be important for the summary. Some constant weight is added while weight calculation.

**Sentence Length[19]**: Number of words in a sentence defines the length of sentence,which is a factor in deciding sentence relevance in the final summary. Medium length sentences are more suitable to be included in the summary. As short sentences are assumed to include less information and Long ones are assumed to depict the detailed analysis.

**Proper noun[19]:** Source document sentences which include proper nouns are assumed to be suitable for the final summary. Name of a person, place, group or a thing are examples of a proper noun.

**Proximity[19]**: To identify relations among words or entities, distance between them is considered an important factor.

**Similarity**: To find out the relevance of a sentence in a document we calculate similarity among sentence and other sentences of the source document. Character string overlap and Linguistic knowledge can be used to find out the similarity.

The summarization work was first initiated at the University of Columbia by a Natural Language Processing group; a system named SUMMONS[15] was developed for summary generation. Initially, they came across with different approaches and challenges but afterward new perspectives were provided after association with people of different communities. Some approaches use clustering to find out key themes and extracting one key information unit from each cluster. Some approaches extracted one and other extracted more than one as per their need. Some used maximum margin relevance to include a passage by analyzing the previous information.

Figure 2.1 discusses the taxonomy of automatic text summarization and this chapter further discusses the phases of this taxonomy.

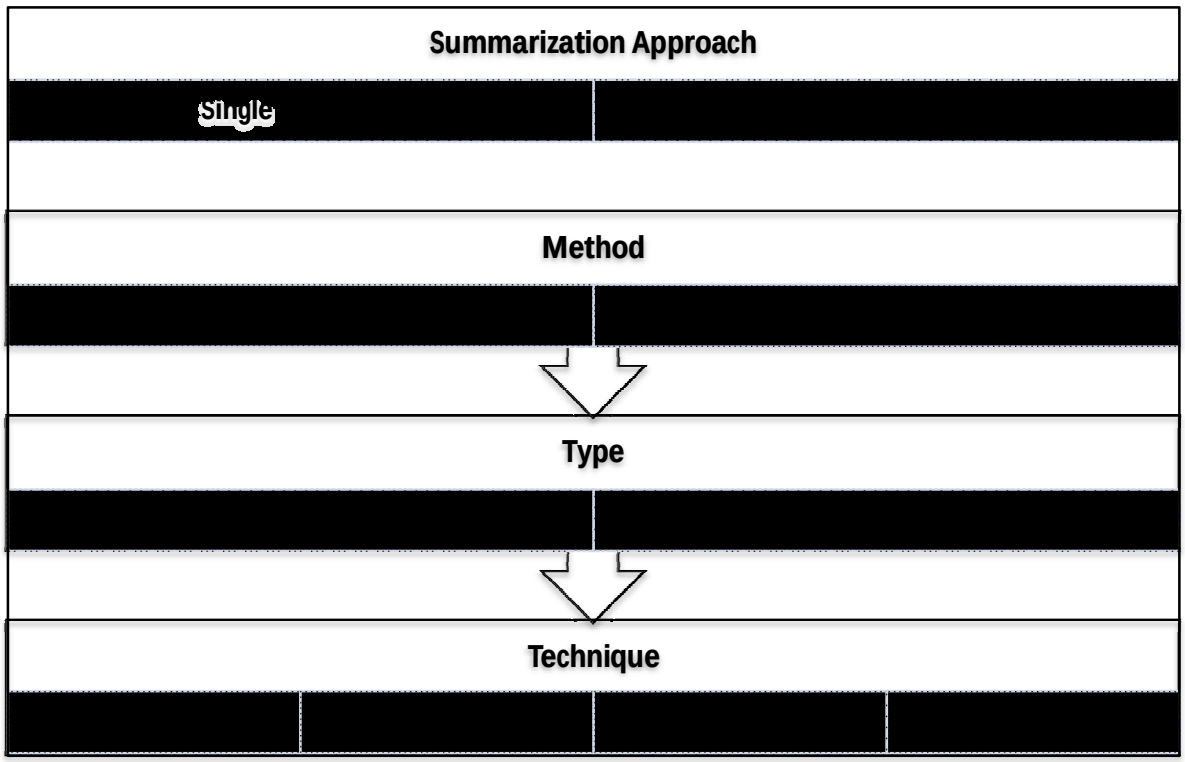


**Figure 2.1: Taxonomy of Automatic Text Summarization []**

## 2.2 Summarization Approaches

The summarization can be performed on single documents and multiple documents as well.

**Single Document Summarization**

In single document summarization[16], one source document is analyzed and processed to generate a quality summary. It is a simple and most probably first approach towards summarization. Both the approaches, extractive as well as abstractive can be applied on a single text document.

**Multi Document Summarization**

Multi document summarization is a technique which involves the information extraction from more than one document. Multiple source documents are analyzed and evaluated to generate an important and non-redundant piece of information. Processing multiple text files is a difficult and tedious task to perform as compared to process single document. Multiple document summarization[17] technique came after single document summarization to cater the needs when we need to concise data which is distributed in multiple files. News on the internet is based on the web based clustering systems. News articles are generally published on the websites after combining summaries from multiple sources after removing redundant and unnecessary content. Output summary produced from this technique should

Be coherent and complete in itself. To generate and assure that a summarizer produces coherent summary we need to include linguistic methods.

## 2.3 Methods

Two methods as shown in Figure 2.1, extractive and abstractive are discussed here.

**Extractive Summaries**

Extractive summaries are simple to form as they only include few important sentences from the text document. They decide the importance of sentences in a document and decides to include the most informative sentences, paragraphs etc in the final summarized result. The selection of the most informative sentences is done regarding the features like statistical, linguistic features.

**Abstractive Summaries**

Abstractive summaries are prepared with a combination of newly formed sentences by analyzing a set of important information. New formed sentence should be coherent and complete. Abstractive summaries are generated by proper understanding the source document and then forming new sentences. It produces a representation of internal semantic details, then uses the natural language techniques for the final summary generation. Abstractive summaries may include synonyms or a new set of words to produce an understandable, coherent and informative summary.

## 2.4 Types

Query based summaries and generic summaries are the two types of summarization types.

**Query based Summaries**

In Query based summaries[18], the final summary is generated on the basis of query raised by a user. This technique can be applied on the single document as well as multiple documents. The relevance of a sentence for the final summary result is calculated based upon the frequency of words in a document. A sentence in the original document which includes the keywords provided in a query by the user is scored high than others. Sentences with the high scores are suitable for final summary. New sentences can also be formed by combining information from multiple sentences.

**Generic Summaries**

Generic summaries provide a complete review of the source document unlike query based technique only caters to the query of the user. For the content overview, generic summaries are suitable. This aims to identify the key topics and decrease the redundancy to a possible minimum. Generic summaries categorize and describe the main idea of the source content.

## 2.5 Summary Techniques

There are four summary techniques which are described as follows.

- Semantic and Syntactic (Rule-based)
- Statistical Technique
- Clustering Technique
- Machine Learning Technique

### 2.5.1 Semantic and Syntactic (Rule-based)

The Semantic and Syntactic analysis is used to find and present the association among different sentences by applying on source content for text summarization. Three Semantic and Syntactic summary techniques are:

- Graph Representation
- Lexical Chains
- NLP (Natural Language Processing)

The graph representation is done during summarization by lexical graphs, sentences are represented as Weighted graphs, unweighted graphs, graph matching etc are tasks performed during summarization process.

Lexical chains are used for building chains of identified units for summarization with the help of co-reference chains and lexical semantics etc.

Natural Language information processes language data to extract information also uses part of speech for summary production. There are two techniques for summarization under Natural Language Processing listed below:

1) Plain text Summarization
2) Multilingual Summarization

In Plain text summaries resultant summary is in the same natural language but in multilingual text summarization[21] resultant summary is in different natural language. Initial work in plain text summarization was started in 1950's. Most initial work on the text summarization was targeted on technical documents. Luhn (1958) proposed the first algorithm for text summarization at IBM. The author proposed text summarizer which was based upon the frequency of a particular word in a document. The main motivation was to shorten the news information, biographical information. According to the Luhn, the summary has different categories, some of the summaries are difficult to generate than other. Different categories are Extractive, Abstractive, Indicative, Informative and Critical. Extractive summaries are simplest. These summaries contain the sentences which have already presented in the text. Abstractive summaries contain some new text also. Indicative summaries represent the scope of the whole document without including whole content. Informative summaries represent the important factual content of the text document. Critical summaries represent reviews on scientific papers

about their work and results. Baxendale (1958) also did work related to extractive summaries at IBM. The author more focused on the “sentence position”. Approximately 200 documents are analyzed for research. Edmundson [22] proposed the system for document extraction. The proposed algorithm was the first algorithm for extractive summaries. Two previous features sentence frequency and position of the sentence were used along with the new features like cue words and skeleton. Cue words are words like hard, significant etc. Skeleton define the heading of the document. After evaluation 44% results matched with the manual results.

Multilingual text summarization came into existence in 2005. This technique is still in early stage but this different framework has many advantages in the Newswire field in which information is combined from different foreign news agencies. Evans (2005) described the scenario in which there is always a preferred language in which summary is required, different multiple source documents are in demand and in different languages are available. They preferred English as a source language and documents are from the news articles in English language and Arabic. The logic was to generate the summary of English articles without discarding the details contains in Arabic. IBM’s machine is used to do a transformation of Arabic language to English. The system checks the transformed document in Arabic corresponding to a document of English for each sentence. If a match is found then the sentence is found relevant for summary. Hence more grammatical summary is found since machine translation is still not perfect of that. To find out the similarities between sentences Simfinder tool was used. This is a clustering based tool based upon similarity over different semantic and lexical features which is using long linear regression model. Universal Networking Language is mostly used in multilingual summarization.

Martins and Rino[23] proposed an algorithm for the text summarization using UNL. They presented UNLSumm model to prune the UNL text by means of heuristics that totally focus upon unnecessary binary relations. The system used the decoder to produce the corresponding summary in Brazilian Portuguese. Their pruning heuristics are based upon the relations of UNL. Although each relation is not a candidate for pruning because some relations like “agt” or “obj” convey important information. Only some of the relations are candidates for pruning. According to this algorithm initially, there was 84 heuristics were divided into two groups A and B shown in Figure 2.2 and Figure 2.3. Group A considers 39 heuristics. It also called as single pruning and removes the independent binary relations one by one. Group B heuristics are complex than the

Group A heuristics. Group B heuristics are called chained pruning, i.e., once the binary relation is excluded the interconnected binary relation is also excluded.

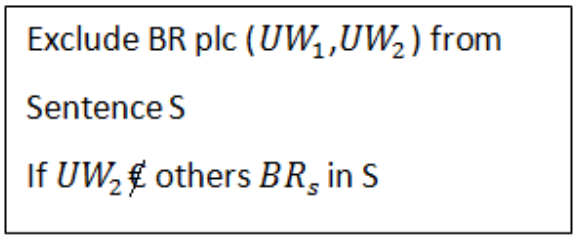


**Figure 2.2: Group A heuristics**

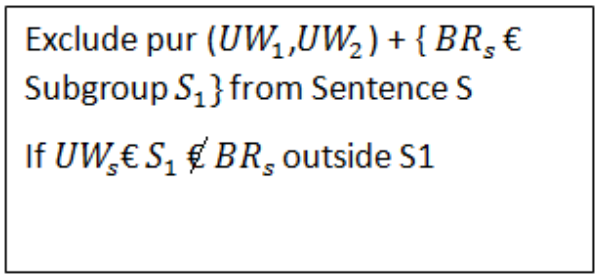


**Figure 2.3: Group B heuristics**

According to Figure 2.2 Group A heuristic deletes the place relation from the UNL document, provided the frequency of UW2 is one means UW2 should not be a part of any other relation in the same UNL sentence. While applying Group B heuristics frequency of UW2 should be 2. These heuristics are more complicated because deleting the desired relation containing UW1, UW2 leaves blank in any other relation where UW2 is placed. For example, if purpose relation as shown in Figure 2.3 is deleted containing UW2 then any other relation in same UNL sentence containing UW2 no more will the part of UNL document. The serious problem regarding these heuristics is to decide the heuristics application order when considering both types of pruning. By default Group A heuristics are always applied and in the case of inter-dependency when dangling of binary relations occurs, Group B is applied. However, Group A and B work on the same binary relations but sometimes after applying Group B heuristic results into more than one dangling relations. Hence, to give a priority to Group A or Group B heuristics is one of the major issues. The precision of the Heuristics is calculated represented by this formula:

**Precision (H) = Sat_num/ Total_num**

Sat_num= Number of application of H Leading to Satisfactory Results

Total_num= Total number including Satisfactory and Unsatisfactory Results

There are some limitations of this approach which are as follows.

Sometimes it covers non-relevant information. There is an upper bound to the number of heuristics applied for each entry. Application order is relevant and providing satisfactory results or not.

### 2.5.2 Statistical Technique

For extraction of relevant information, some systems use Statistical Techniques. This technique uses statistical methods generally applied with Binomial Distribution, sentence compression and calculated scores. This technique is used by Hidden Markov model.

Conroy and O'Leary [24] employed statistical technique by hidden Markov model[] approach for summarization of plain text documents. A sequential Model was prepared for the evaluation of local independence.

This system has three key parameters as the length of the sentence in processing, the position of the sentence in document and likeliness of key terms in the sentence being evaluated.

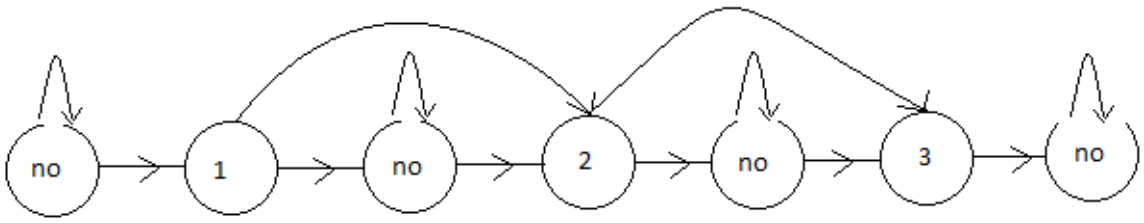


**Figure 2.4**: *Markov model to extract the three summary sentences* [23]

Figure 2.4 represents Markov model where the circle represents summary states and non-summary states. Here, Circles inscribed with a "no" are displaying the non-summary states and circle inscribed with numerical numbers like '1', '2' etc represents the summary states. There is no possible jump from summary states to next state whether next state is a summary or non-

summary state. Here, Figure 2.4 represents the model with 7 nodes corresponds to three summary states, s=3, and four non-summary states.

### 2.5.3 Clustering Technique

When multiple objects are grouped together based upon their properties and characteristics, this process is termed as Clustering[6]. A cluster consists of the objects having similar properties. In text summarization, we use clustering to group similar type of sentences together. In a document different topics are arranged in a specific ordering. In Cluster based technique, sentence selection is done after cluster generation. The sentence is also chosen based upon the location or position of the sentence in a document. A score of a sentence increases if it has multiple occurrence hence higher probability of selection in the final summary.

Overall Score of a sentence is evaluated to check the relevance of the sentence ***Si*** is:

$$S_i = \text{W1} * C_i + \text{W2} * F_i + \text{W3} * L_i$$

In the above formula, is total score of the sentence, Ci is a cluster to which sentence belongs, is a document to which sentence belongs and Li is the location of a particular sentence.W1, W2 and W3 represent the weights.

### 2.5.4 Machine Learning Technique

Machine learning techniques are very effective for automatic text summarization. The some of the machine learning approaches are discussed as follows.

**a) Naive Bayes Approach**

Kupiec (25) described a method for summarization. He described a classification function known as naïve Bayes classifier which is responsible for the each sentence to be a part of the summary. If denotes the total number of sentences and s denotes a particular sentence with features [16]. The formula of naïve Bayes is shown below:

$$P(s \in S|F1, F2, \ldots . Fk) = \frac{\left(\pi_1^k P(F_i s \in S).P(s \in S)\right)}{\pi_1^k P(F_i)}$$

The new features like sentence length and the upper case words were added. The Score is calculated for each sentence and based upon that top most n sentences were chosen. Aone et al. [26] also describe a naïve Bayes classifier with more additional features. He introduced the terms “frequency” and “inverse document frequency” in plain text summaries. The corpus used in the experimental analysis was from the newswire. The inverse document frequency was computed from a large corpus of the same area.

**b) Rich features and Decision Trees**

Lin and Hovy (27) describe the importance of a feature “sentence position”[]. According to this, a weight is provided to sentence based upon its position in the text. This method also called as position method. A newswire corpus was used for experimental analysis. The authors measured the yield of every sentence position. They ranked the different sentence positions to produce the “Optimal Position Policy (OPP). They performed the two kinds of evaluations. They test on the unseen text. The first evaluation was exactly like the training documents and the second evaluation considered the word overlap for the manual abstracts was measured. Abstract windows and selected sentence windows were compared and precision, recall values were measured. Lin (1999) broke away the assumption that the features are independent and tried to model the problem using decision trees instead of the naïve-Bayes classifier. The system described a lot of features in sentence extraction and their effects. The data set used was publicly available texts classified into various topics. The data set is divided into text fragments which are evaluated by human judges. Some important features were query signature (normalized score of the sentences depending on the number of query words), IR Signature (the salient word like the signature word), numerical data, proper name (Boolean value 1 is given to sentence that had a proper name), pronoun or adjective (Boolean value 1 is given if they appeared), weekday or month, Quotation, query and signature. The system experimented with different baselines like positional feature, simple combination of features. When machine extracted and human extracted sentences were matched, the decision tree was clearly the winner.

**c) Log Linear Models**

Osbrone (2002) described the Log Linear model approach[28] for the plain text summarization. This approach is different than the previous approaches which always

assumed feature independence. The system showed that this approach is better than naïve Bayes classifier approach. The model can be stated as follows:

$$P(c|s) = \frac{1}{Z(s)} \exp\left( \Sigma_i \, \lambda_i F_i(c,s) \right)$$

Where Z(S) is

$$Z(s) = \sum c \, (\sum i \, \lambda_i F_i(c,s))$$

In these equations c is a label, s is an item to be labeled, $F_i$ is a feature (i- th feature) and λ is weight of the feature. There are two possible labels regarding whether the sentence is to be extracted from the document or not. The weights given to sentences are calculated from conjugate gradient descent. The non uniform prior is added to the model by authors. This model rejects too many sentences during processing. The features included by the authors were word paring, length and position of the sentence and discourse features like inside the introduction, part of conclusion.

**d) Neural Networks**

Svore (29) produced an algorithm based upon neural networks and used the third party features like dataset to resolve the problem of extractive summarization. The data set consists of 1365 documents collected from CNN.com. The datasets consist of human generated stories, articles, title, and timestamp etc. For the evaluation, two metrics were considered. The first one is to combine the system produced three highlights, combine the human generated three highlights and comparison of these two. The second take care about the ordering and the individual level comparison of the sentences.

Strove (2007) trained this model on the basis of labels and featured for each sentence that referred the ranking of each sentence in the source document. The Ranking was provided to the sentences on the basis of RankNet which was a paired based neural network algorithm. ROUGE-1 is used as a training set. The authors concluded that if a sentence contains keywords regarding new search engines and Wikipedia articles then the probability of a sentence in highlight is more.

**e) Other Approaches**

**Deep Natural Language Analysis Methods**

Barzilay and Elhadad (30) also described the technique for summarization. It is called Deep NLP analysis. The system described lexical chain that is formed using sequenced words in a given text, neighbor words called as spanning short and long distances. The following steps were used by them. First of all segmentation of the whole text, lexical chains identification, strong lexical chains are used for identification of the sentences. The system described cohesion in the document means togetherness of the different parts of the text. In the lexical cohesion semantically related words are used. For example, consider the sentence:

*Rohan visited Bangalore. He loved the city.*

In above sentence, the word "city" is referring to the word in the preceding sentence "Bangalore". It represents lexical cohesion. The cohesion phenomenon occurs at word level as well as sentence level too. This results into lexical chains which are building blocks for summarization. Relation of the different words and their sequence was also found out which result into several chains and responsible for document representation. For the lexical chain determination Word net was used and three steps were applied.

1. Selection of candidate word set.
2. For each candidate word, find an appropriate chain.
3. Word is inserted in a chain (if found) and then further updating is carried out.

## 2.6 Comparative Study

As Text summarization first approach came in the 1950s since then many new approaches and techniques have been implemented and exercised. Different techniques involve a specific set of feature selection and the content on which the algorithm is applied, We have done a compared analysis of few techniques in the table listed below:

***Table 2.1**: A Comparative Study on Text Summarization Methods Based On Method, Features and Content Selection, Technique used And Summarization Approach.*

| **Author/Year** | **Method** | **Features/ Content Selection** | **Technique Used** | **Summarization Approach** |
|---|---|---|---|---|
| 1995 Julian Kupiec[25] | algebraic method | like length, the position of words, uppercase words | using a naïve-bayes classifier | Extractive Summarization |
| 1997 ChinYew Lin[41] | algebraic method | the position of sentences | Rich Features and decision trees | Extractive Summarization |
| 1999 Eduard Hovy[31] | symbolic word knowledge | concepts relevancy | NLP processing | Single Document Summarization(A) |
| 2005 S.P Yong[42] | Text pre-processing and subsystem | Keywords Extraction Summary production | used neural networks | Abstractive Summarization |
| 1984 Ruqaiya Hasan[43] | Coherence relation | similarity chains | lexical cohesion | Single Document Summarization(A) |
| 1988 William C.Mann [44] | Tree based | to encode the terminal nodes of a tree | RST (rhetorical structure theory) | Abstractive Summarization |
| 1997 Branimir Boguraev[45] | Saliency based content characterization | rank the important sentences | Ranking algorithm | Extractive Summarization |
| 2010 Li Chengcheng [46] | rhetoric relations | candidate sentence | RST (rhetorical structure theory) | Abstractive Summarization |

| | | | | |
|---|---|---|---|---|
| Xiaojun Wan in 2008 [47] used graph based method by introducing | used graph based method | The two-link graph for both sentences and documents | Graph based method | Multiple Document Summarization(A) |
| 2012 Tiedan Zhu [48] | Sentence closeness Parameter | Logical closeness to document | Sentence Co-relation Method | Multiple Document Summarization€ |

This table gives us a review of the text summarization techniques and methods used over the years. Different types of features have been used for content selection with different methods to produce a better quality summary. Naïve based classifier, Graph based method, Sentence co-relation, ranking algorithms are examples of the various method used to get content rich and unambiguous summary.

# Chapter 3

# Single Document Summarization Algorithms

This chapter discusses the different text summarization algorithms which summarize single documents. Each of these algorithms is explained briefly. These algorithms are then implemented and compared for chosen datasets.

## 3.1 Text Summarization Algorithms

Text summarization is done to shorten the text and get to the main point of the document. Summaries are easy to read and understand. Following text summarization algorithms are studied in this thesis.

1. TextRank
2. TextTeaser
3. Summary using Word features

### 3.1.1 TextRank

TextRank[5], an unsupervised algorithm based on weighted-graphs from a paper by Mihalcea et al. It is built on top of the popular Page Rank algorithm that Google used for ranking web pages. TextRank works as follows:

1. Pre-process the text: remove stop words and stem the remaining words.
2. Create a graph where vertices are sentences.
3. Connect every sentence to every other sentence by an edge. The weight of the edge is how similar the two sentences are.
4. Run the PageRank algorithm on the graph.
5. Pick the vertices(sentences) with the highest PageRank score

In original TextRank the weight of an edge between two sentences is the percentage of words appearing in both of them. This TextRank uses a function to see how similar the sentences are.

Graph based algorithms are used to rank the text sentences or words for summarization. To enable working with text on these algorithms, text is represented as graph, where a word depicts the nodes of the graph and edges represent meaningful relations among nodes. Edges represent the connection between two vertices of the graph. Sentences or collocations may also be assigned as vertices of the graph depending upon the size of input dataset. Edges may represent lexical relations, content overlap etc.

**Keyword Extraction:**

Keyword is a method to locate the main keywords in the document which represent the subject of the present information.These identified words contains the most relevant content of the document. An automatic index of a document collection may be prepared by accumulating lists of these words. Keyword extraction can be efficiently used in making dictionaries associated to specific domains. Keywords chosen by this method are present in the original text. Formation of new words or similar words is not considered as Keyword extraction. Selected keywords represent useful entries for information retrieval, data mining and text summary generation. A very simple approach to identify significant keywords is by calculating term frequency. Others may include popularity, context, position etc. to find out the key phrases.

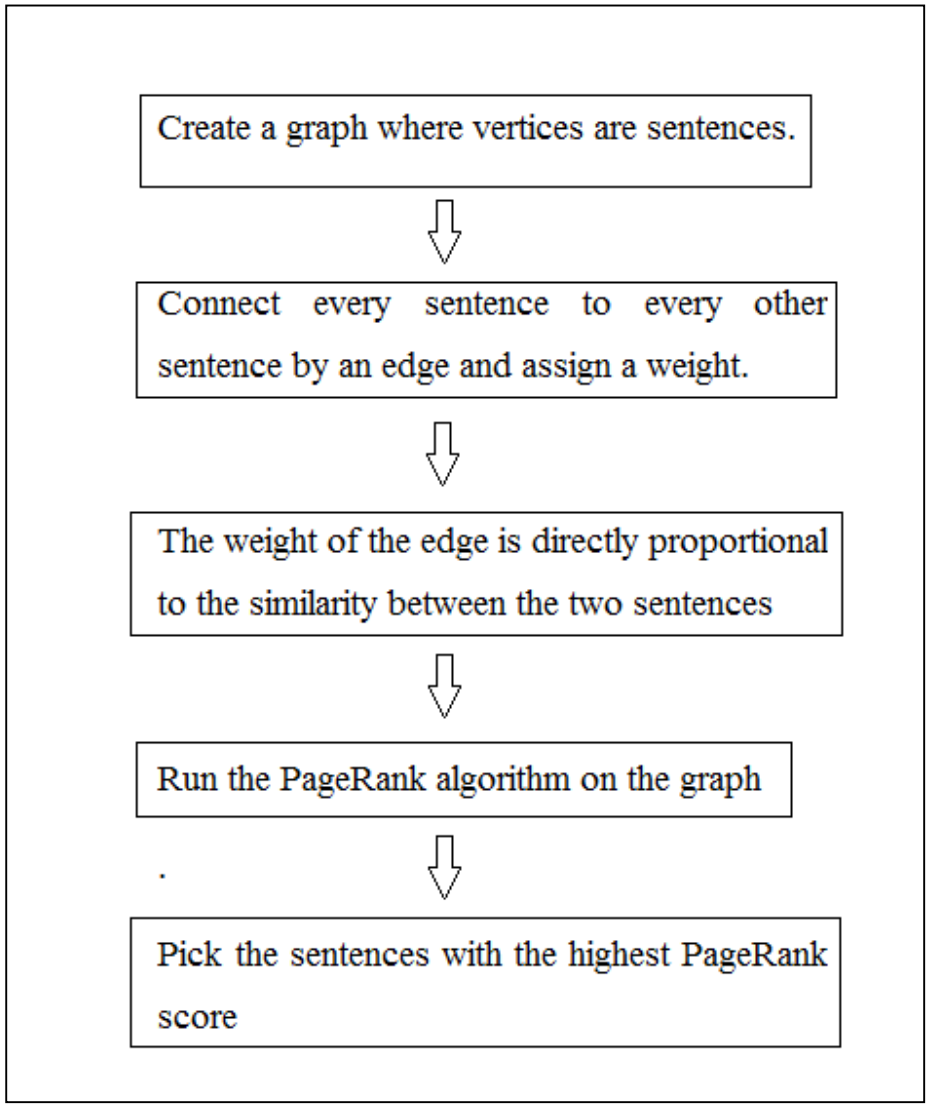


**Figure 3.1: Flow chart of TextRank[14]**

PageRank[14] presents a popular method to calculate the importance of a page in a set of pages joined together by links. It works by measuring the quanitative and qualitative score of links associated to a specific page. It computes an approximate score on the basis of that more websites are likely to contain forward links to important and popular websites. This algorithm analyse the links among different pages and assigns a numerical score to each element of the connected document set. It measures the relative importance of an entity in a set, like in the World Wide Web. The PageRank algorithm can be used to evaluate importance for any collection of elements which has references among themselves. For an element D, P(D) represents the associated PageRank.

## 3.2 TextTeaser

TextTeaser is based upon sentence features[16], which is a heuristic approach for extractive text summarization.

TextTeaser associates a score with every sentence. This score is a linear combination of features extracted from that sentence. Features that TextTeaser looks at are:

- *titleFeature*: The count of words which are common to the title of the document and sentence.
- *sentenceLength*: Authors of TextTeaser defined a constant "ideal" (with value 20), which represents the ideal length of the summary, in terms of a number of words. *sentenceLength* is calculated as a normalized distance from this value.
- *sentencePosition*: Normalized sentence number (position in the list of sentences). Introduction and conclusion will have a higher score for this feature.
- *keywordFrequency*: Term frequency in the bag-of-words model (after removing stop words). Keyword frequency is just the frequency of the words used in the whole text.

More on the sentence features for summarization see Sentence Extraction Based Single Document Summarization by Jagadeesh et al [16].

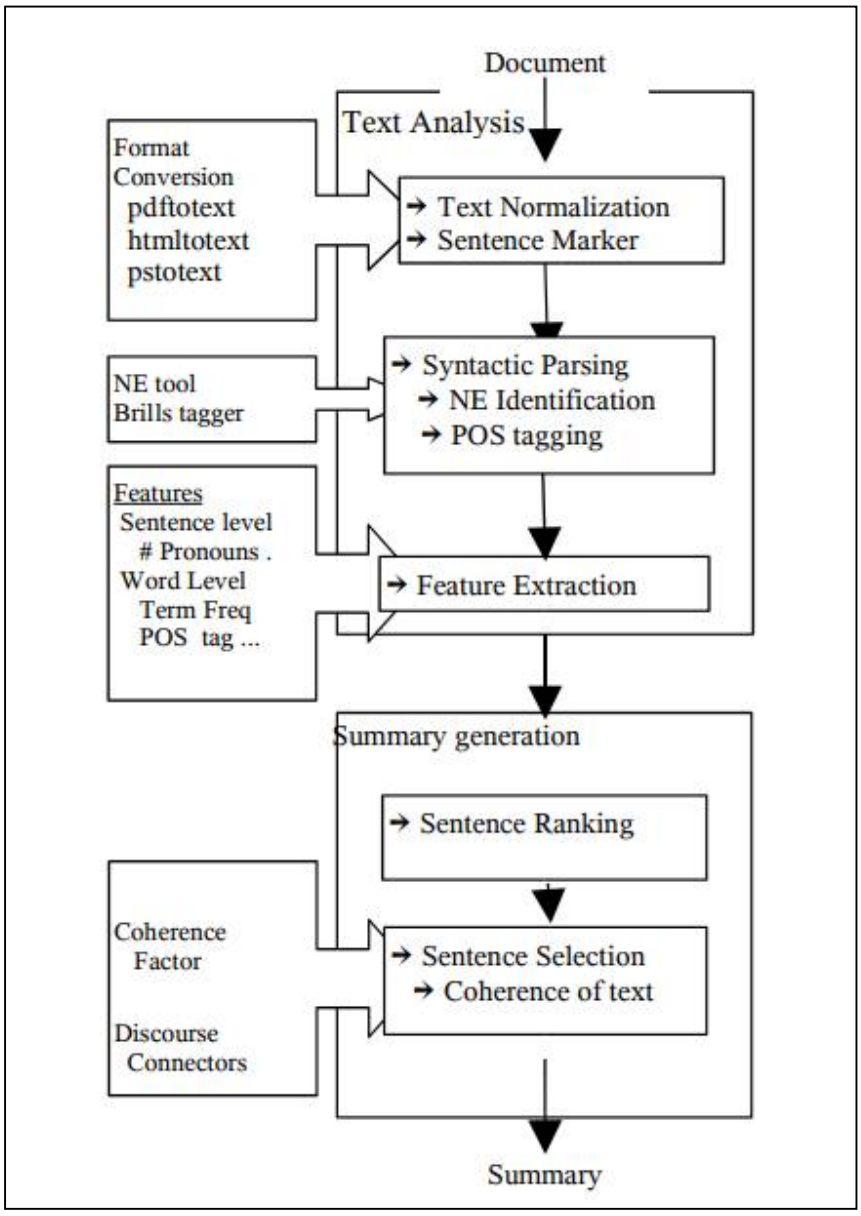


**Figure 3.2: Flowchart of TextTeaser[16]**

Sentence Marker: It is used to split the document into sentence units.

Syntactic Parsing: It is done by sentence structure analysis using NLP tools like Brills tagger [Brill], named entity extractor, etc. This extractor recognizes named entities ( like persons, organizations, and locations etc), temporal expressions (time and date) and specific numerical values expression from textual data.

Feature Extraction: Both the word level features are extracted to be used in the calculation of the relevance and importance of the sentence present in the document. The word level features are listed below:

1. Length of the word l(w)
2. Familiarity of the word f(w)
3. Parts of speech tag p(w)
4. Term frequency tf(w)
5. Font style F(**w)**
6. Occurrence in headings O(w)

The sentence level features are:

1. Length of the sentence
2. Presence of the verb
3. Pronouns referring to preceding sentences
4. Position of the sentence in source document

**Sentence Ranking and Summary Generation**

Most of the times word features depends on the context of its occurrence, i.e they may depend on the sentence position and number also(ex. POS tag, familiarity, ..). Similarly, the word score also depends on the sentence number in the document. Once the feature vector is extracted for each sentence, the score of a sentence is calculated by obtaining the total sum of individual words as :

$$\text{Score (l , w)} = \prod_i fi(w)$$

$$\text{Score}(l) = \sum \text{Score}(l, wi)$$

where l, represents the sentence number and ‘w’ represents the word present in the sentence, and f i (w) represents the ith feature value.

After the sentence scores are assigned, sentences are selected to form good summary. One method is to extract the top N sentences but this may lead to the coherence problem.

**Coherence Score(** CS): Coherence score[32] is used to identify the amount of common information between the set of already selected sentences and the new sentence to be included. A list of words is used to evaluate the coherence of the sentences.

Let $S_w$ represents the set of words in the already selected sentences, and $l_w$ denotes the set of words present in the new sentence to be selected, then coherence score is obtained by the total sum of the common word scores. Now the score of the new sentence is computed by

$$CF \times CS(l) + (1 - CF) \times SPW(l)$$

where CF denotes the Coherence Factor.

## 3.3 Summary Algorithm based on Word Features

This Algorithm[33] aims to provide an efficient manner of reducing a document to an understandable text, which is done by selecting the most important sentences. The algorithm has following key steps:

- Ranking sentences using the below-described algorithm.
- Transition phrases and unnecessary clauses are removed.
- Excessive examples are removed.
- Reorganizing the summary to focus on a topic; by selection of a keyword.

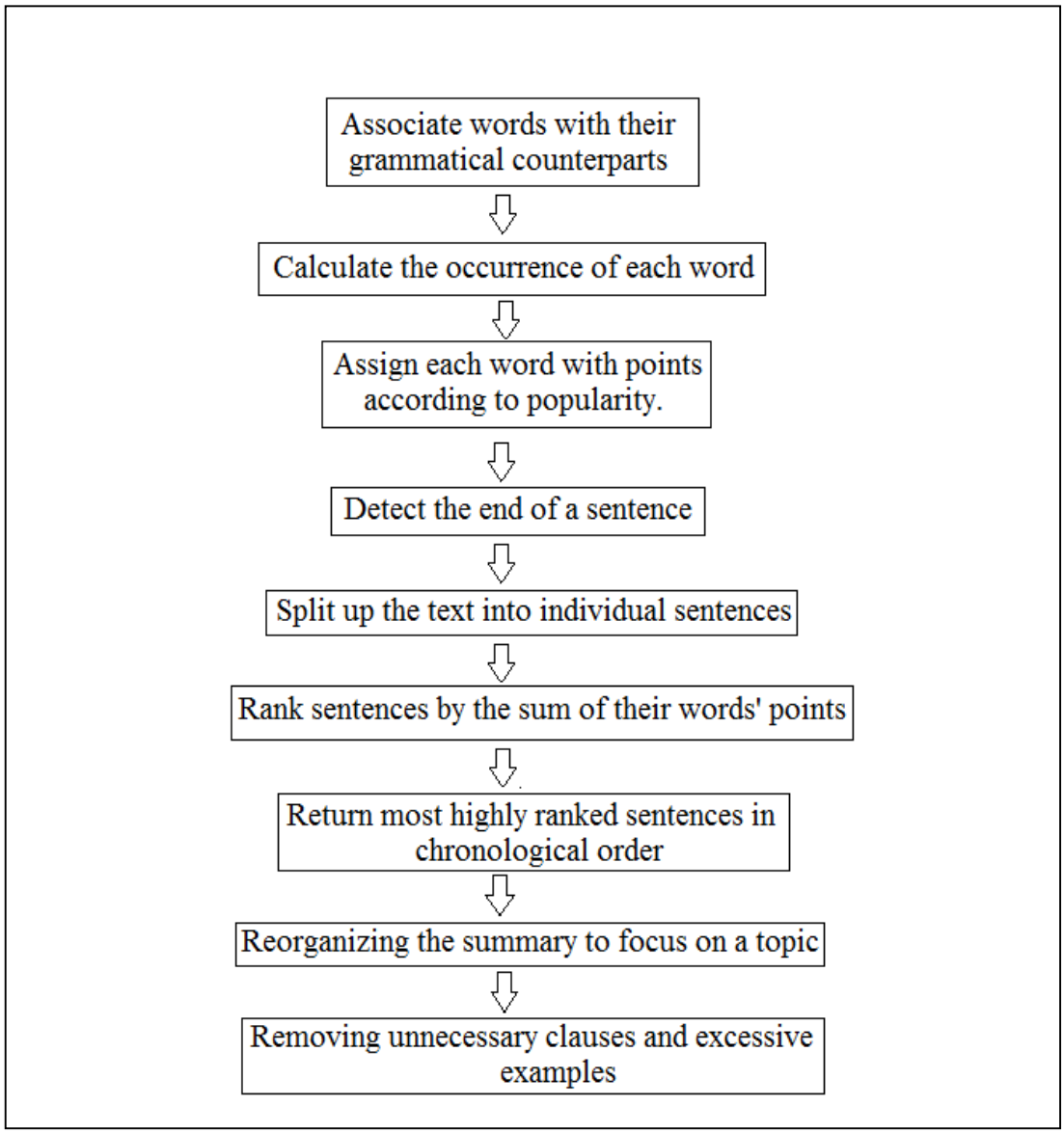


**Figure 3.3: Flowchart of Summary Algorithm**

The core algorithm has 7 key steps listed below:

1. Associate each word with the grammatical equivalents. (e.g. "light" and "lights")
2. Compute the frequency of each word in the document.
3. Assign each word with points depending on their popularity.
4. Determine the correct ending of a sentence. (e.g "4.5" does not).
5. Separate individual sentences from the text.
6. Rank sentences on the basis of obtained sum of associated words' points.
7. Select X topmost sentences.

# Chapter 4

# Multi-document Algorithms

**Multi-document summarization** is an automatic procedure to create a summary which includes important information on key topics from multiple documents. It creates a concise and comprehensive summary. Here, Algorithms are presented to perform summarization based on different methods to evaluate the accuracy of produced summary for different Multi-document datasets.

This Subsection includes various Multi-document text Summarization algorithms based on:

- Summarization Using ILP Based Multi-Sentence Compression
- LDA topic model
- Sentence Clustering

## 4.1 Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression

Abstractive summarization is an ideal form of summarization as it alters the given source documents sentences to form new informative, non-redundant and coherent sentences to be included in the final summary. Sentences produced should be easily understandable and readable. To form completely new phrases matching to the human understanding is not yet achieved but this algorithm tries to maximize information content by combining words from multiple sentences.

This Algorithm performs Multiple document summarization using integer linear programming model[34] which aims to produce coherent and highly informative sentences. First, Algorithm employs LexRank[35] to find out the most important document from the set of source documents. Then, the sentences belonging to the most important document are aligned to the sentences of another document to generate clusters of similar sentences. In each of the generated cluster, k-shortest paths from the sentences are generated with the help of word-graph structure. Finally, sentences are selected by the help of shortest paths generated employing a novel integer

linear programming method in order to form new informative coherent sentences. Above stated shortest paths are represented as binary variables in the ILP method and number of words in a sentence path, information and quality score are considered in the function.

**Steps in the Algorithm:**

There are two main steps in the algorithm:

1. Sentence Clustering

2. Summary Sentence Generation

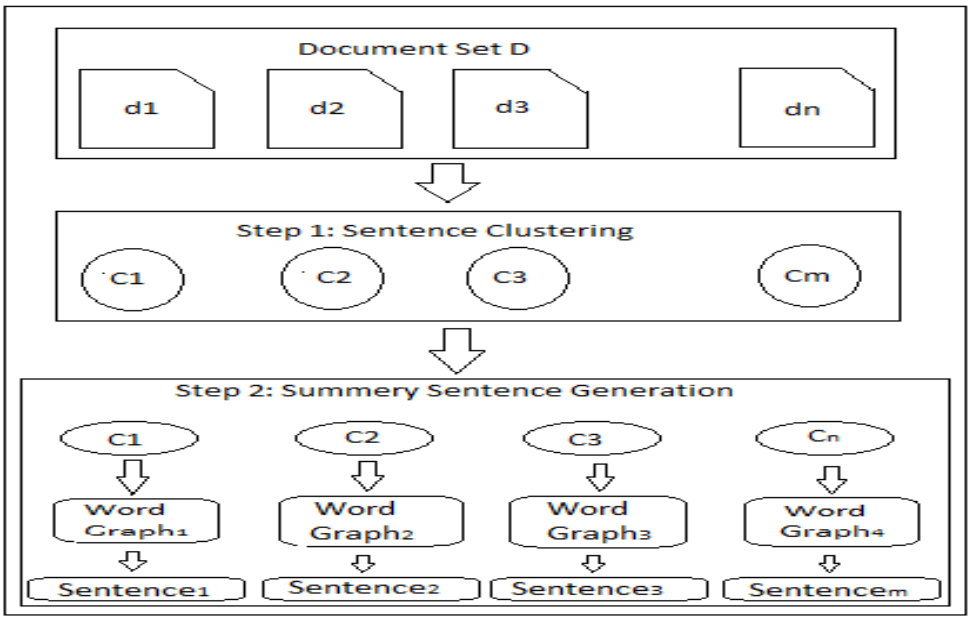


**Figure 4.1 Overview of the Summarization approach**

The above stated two steps of the Algorithm are further divided into the following steps:

Step 1: Sentence Clustering

Clusters of sentences are created using each sentence from the most important document, $D_{imp}$, in a document set D. It is assumed that $D_{imp}$ is comprised of the most important content across all the documents present in the set D. The document which contains more similar information to central content is most informative.

(Step 1.1) Document Importance

We propose several techniques to identify $D_{imp}$.

**LexRank:** LexRank [35] creates a sentence graph where the edges represent weights which are calculated by the help of inter-sentence cosine similarities. While in this algorithm, a graph of documents is constructed to calculate the importance of a document. The equation below shows a formula to calculate LexRank score for a node in a graph using weighted links present among nodes. This computed score represents the importance of the document in the set of input documents. Let p(x) denotes the centrality of node x in the equation below:

$$p(x) = \frac{d}{n} + (1-d)\sum_{v \in adj[x]} \frac{\text{idf-modified-cosine(u,v)}}{\sum_{z \in adj[v]} \text{idf-modified-cosine(z,v)}}\, p(v)$$

where adj[x] denotes the set of adjacent nodes to u and N represents the total number of nodes present in the graph, 'd' denotes damping factor(set to 0.85). Document representing the node with the highest LexRank score is a most important document, $D_{imp}$ for the set of input documents.

Pair-wise Cosine Similarity: It is used to calculate the average cosine similarity between the current document $d_i$ and the other documents present in the input dataset. The equation to find out average cosine similarity is:

$$\text{AveCosSim}(d_i) = \frac{\sum_{di,dj \in D} CosSim(di,dj)}{|D|-1} \qquad (i \neq j)$$

where [D] denotes the number of total documents present in the document set D.

Overall document collection similarity: This method is used to calculate the cosine similarity between the current document($d_i$ ) and the whole input document set. We obtain the document set by concatenating the data from all the documents of the dataset D. This is calculated as:

$$DocSetSim(di) = CosSim(di,D):$$

After selecting the most important document, $d_i$ from the input dataset D, we create the clusters by aligning sentences and arranging them based on their original positions in the input documents.

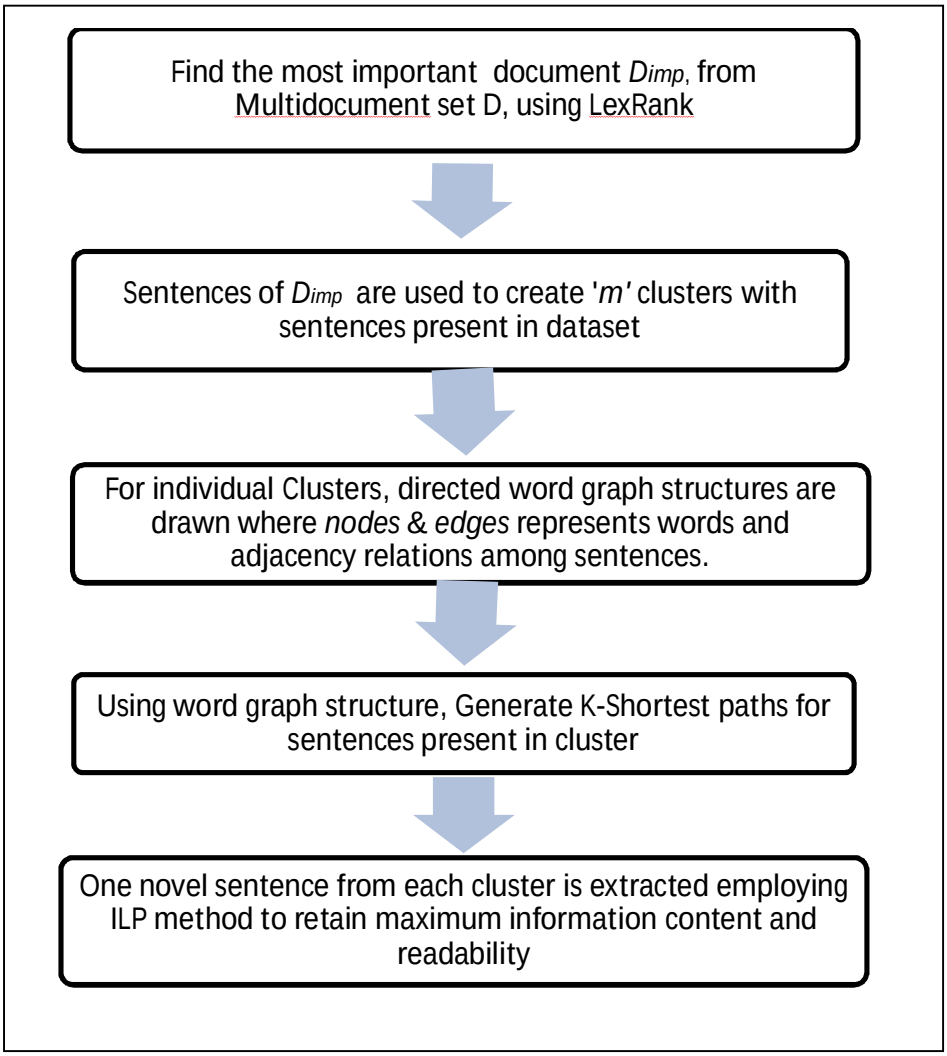


**Figure 4.2: Flowchart for Multidocument Abstractive Summarization using ILP based Sentence Compression**

## 4.2 Multi Document Summarization Algorithm based on LDA Topic Model

This Multi Document Summarization Algorithm[36] is based on the Latent Dirichlet Allocation (LDA) topic model which takes a multiple numbers of documents as input and generates a final output summary including an important piece of information from all the input documents. Latent Dirichlet allocation is a popular topic model which finds topics on the basis of word frequency i.e. occurrences of a word from a set of input documents. It presents the input text as a mixture of latent topics; these topics represent the key concepts in the document. LDA is particularly designed for identifying a reasonably accurate number of topics within a given document set. LDA (Latent Dirichlet Allocation) Model is used to find the important topics in the input provided.

These latent topics are useful to employ sentence ranking methods in order to obtain good quality summary. The sentence ranking mechanism calculates the posterior probability of each sentence based on two factors i.e. the topic distribution of the sentence and topic importance. Here, Topic Distribution denotes the degree to which a sentence belongs to a particular identified topic and Topic Importance denotes the importance of the topic depending upon the amount of information covered by this topic in the documents provided. After obtaining the probability for each of the existing sentences, it extracts the important sentences to be included in the final optimized summary based upon the above calculated posterior probability.

**Topic-Importance:** Topic importance represents the significant portion of the document covered by a topic. A topic covering a large amount of content of the document will be assigned higher probability value and vice-versa. All the latent topics identified by LDA will have different probabilities. Topic Importance refers to the posterior probability of the topic in the document. All identified topics in a document are not of equal probabilities, as it depends upon the part of the document which can be represented by a topic. Before selection of sentences to be included in the final summary based upon posterior probability, the topic importance should be calculated.

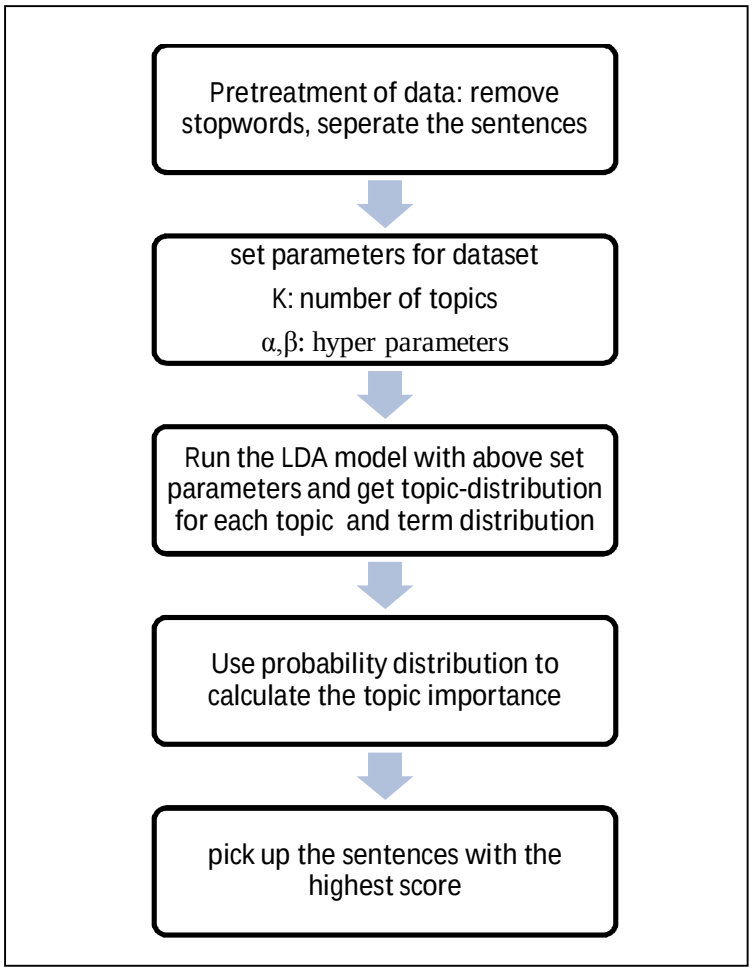


**Figure 4.3: Flow-Chart: Multi-Document Summarization based on LDA topic Model**

Topic importance for a topic distributed among a set of documents is calculated by evaluating the distribution of topic over all the input documents. The prior probability for all the documents is equal, which implies that initial order of documents has no impact on the value of Topic importance.

The formula to calculate Topic Importance is (1)

$$P(T = z_i^d | D) \propto \sum_{d=1}^{D} P( T = z_i^d | D=d)$$

Where $P(T = z_i^d | D)$ refers to the topic importance and $(T = z_i^d | D=d)$ refers to the topic-distribution obtained by LDA Model for a document.

Topic importance is directly proportional to the content covered and the length of the document. Topic covered in a lengthy document has a higher weight assigned in Topic importance. The formula to calculate Topic importance in this case is

$$P(T = z_i^d | D) \propto \frac{\sum_{d=1}^{D}(N_d \times P(T = z_i^d | D=d))}{\sum_{d=1}^{D} N_d}$$

Where $N_d$ is the total number of words in a document $d$ .

**Sentence Ranking Algorithm:**

Sentence ranking is done to select the sentences to be included in the final summary by evaluating the score for each sentence i.e. the posterior probability. The probability depends on two factors, first the topic importance and other is topic distribution. The sentences with high-weight posterior probability are used to form summaries. So, to evaluate the Posterior probability of a sentence, the following method is used.

The degree of a sentence associated with a certain topic is represented by the Conditional Probability. The Conditional Probability is calculated as ()

$$P(S = s_j^d | T= k, D) \propto \prod_{w_i^d \in s_j^d} P(w = w_i^d | T = k, D)$$

Where $P(S = s_j^d | T= k, D)$ represents the conditional probability; $P(w = w_i^d | T = k, D)$ represents the term distribution associated to a topic identified by LDA Model.

The Length of a sentence in a document represents the degree of information it contains. The length of a sentence for information quality is only considered after removing stop words and function words. Product result of calculated probabilities of words may reduce the value to very low for long sentences. So, Product is replaced by summation of calculated probabilities. We get the new formula as :

$$P(S = s_j^d | T= k, D) \propto \sum_{w_i^d \in s_j^d} P(w = w_i^d | T = k, D)$$

For each sentence, the topic-distribution is determined by the joint probability distribution and it is defined by the conditional probability above and the topic-importance as follows:

$$P(\mathrm{S} = s_j^d \mid \mathrm{D}) = \sum_{k=1}^{K} P(\mathrm{S} = s_j^d, \mathrm{T} = k \mid \mathrm{D})$$

Where P(S = $s_j^d$ | D) represents the posterior probability of sentences.

A sentence with lower probability words might have a greater value than a shorter sentence having higher probability words. Thus, the posterior probability of sentences is normalized by the sentence length, we calculate the posterior probability as given below:

$$P(\mathrm{S} = \mathrm{s}_j^d) \propto \frac{\sum_{k=1}^{K} \sum_{w_i^d \in s_j^d} P(\mathrm{w} = w_i^d \mid \mathrm{T} = k) P(\mathrm{T} = k)}{\mathrm{Len}(\mathrm{S} = \mathrm{s}_j^d)}$$

Where Len(S= $s_j^d$) represents length of the sentence.

Based on this Posterior probability, the top sentences are selected for the final Multi-document summary.

## 4.3 Multi-Document Summarization Using Sentence Clustering

This Multi-Document text Summarization algorithm[37] uses clustering technique to extract an important piece of information from input documents. The sentence is considered as the most basic entity while performing Text Summarization. Clustering of sentences, paragraphs or text documents are performed on input dataset to produce a good multi-document summary.

This technique aims to produce Multi-document summary based on Single Document Summarization and sentence Clustering. In the algorithm, Single document summaries are produced by preprocessing and feature extraction of each document present in the dataset. The prepared summaries are combined by semantic based sentence clustering. Important sentences to be selected for final multi-document summary are chosen from these clusters with similar sentences. Non- redundant, coherent and important sentences are extracted for the summary.

The figure below depicts the approach for text summarization of more than one document. As from the figure shown, each input document first undergoes pre-processing and then document

features are selected, which contributes in single document summary generation. The individual summaries are further clustered based on sentence similarity. Newly formed cluster's sentences are selected to prepare the final multi-document summary. The selected sentences are presented in the same order as present in the source document.While cluster generating process of single document summaries, semantic and syntactic similarity among different sentences is considered. The semantic similarity of words of the cluster is added together to get the final similarity score between sentences.

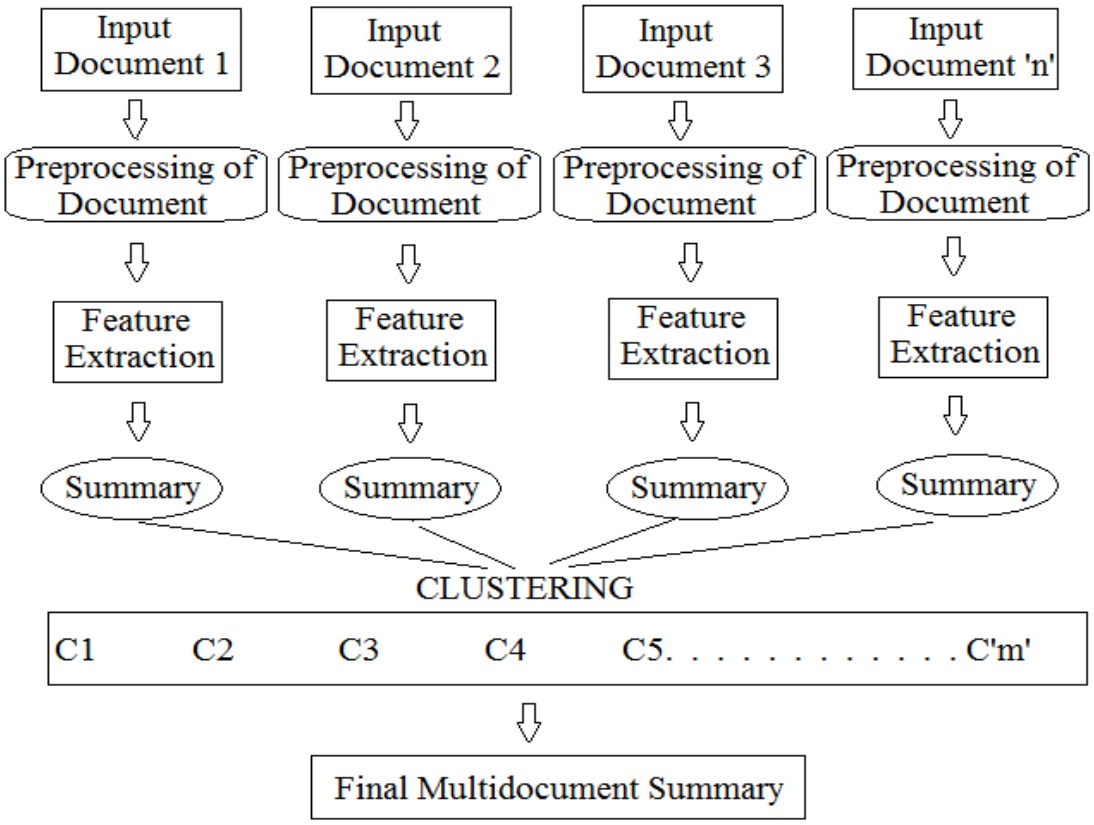


**Figure 4.4: Steps in Multi Document Algorithms**

The significant steps in multi-documents summarization are as follows:

## 1. Pre-processing

Preprocessing of input text plays a significant part in text summarization. Removing stopwords, stemming, separate each sentence by finding coreect end to a sentence and tokenization are few important steps to perform Pre-processing. Stopwords need to be separated from input data as they don't contribute to quality of summary, hence not considered for final summary generation..

Stemming helps in discovering the root of similar words and to decrease the number of morphological variants. For example , the words like summary, summarize, summarization, summaries all are derived from the word 'summary. Various suffixes of the word are removed to reduce ambiguity. Porter Stemmer [38] is used for this algorithm.

To get a Sentence as a unit for different processing, sentence splitting is used which determines where the sentence end. The end markers(. ? !) for sentence splitting may not give desired results in certain cases. Text data like numbers, Abbreviations etc. (7.9, i.e., Ms., Dr., etc.) results in the wrong identification of sentence boundaries. In order to identify correct boundaries simple heuristics and regular expressions are considered. Tokenization explores the text and seperates it intowords, symbols etc.

## 2. Feature extraction

Feature extraction involves representing text data in form of feature sets. Features are properties of existing data which are useful to identify the importance of words and sentences in the document. Here, the following features are extracted:

Document feature: Each sentence in a document is assigned a weight within a document, which is termed as document feature. The weight of a sentence is calculated by adding the weights of all the content words existing in the document.

$$\text{Document Feature(DF)} = w1+w2+\ldots\ldots\ldots\ldots\ldots\ldots wn$$

Here DF represents the document feature of a whole document and $w_i$ is the symbol for normalized weight associated to the $i^{th}$ word of the sentence. For calculating Document feature, words with normalized frequency greater than a certain value are considered.

Sentence reference index (SRI) feature: Sentence containing pronoun represents a reference to preceding sentence. SRI feature increases the weight of the referred sentence. In Order to recognize such a sentence a list of pronouns is prepared.

Location feature: The location of a sentence in a document is considered while weight calculation. Higher weight is assigned to starting and ending sentences and lower weight to middle paragraph's sentences. Top and bottom sentences are assumed to have definition and conclusion of the document.

Concept similarity feature: It is defined by the number of synsets associated with query words simlilarity with the sentence. WordNet[13] is used to get a set of synsets for assigning concept similarity weight to a sentence. For example, WordNet gives below synsets for the unit "dog":

Dog: dog, domestic dog, Canine, Carnivore, Mammal, vertebrate, chordate

## 3. Single document summary generation

The weight of a sentence is evaluated by calculating total sum of individual features as below:

$$SW = v* DF + w* LF + x*SRI + y*CS$$

Where DF is document feature, LF is location feature, SRI is sentence reference index feature, CS is concept similarity feature and alphabets v, w, x, y are constant values. The constant values are fixed experimentally to v=0.5, w = 0.2, x = 0.2 and y = 0.1 are used to compute SW, is a symbol for sentence weight.

The sentence weights are calculated as shown below:

$$\text{Normalized Weight} = \frac{weight\ of\ each\ sentence\ present\ in\ a\ document}{Maximum\ weight\ of\ any\ sentence\ present\ in\ a\ document}$$

Normalized weight is used for ranking of sentences. Top *k* sentences are selected to form single document summary from the source document.

## 4. Multi-document summary generation

The prepared single document summaries are combined together with the help of sentence clustering and then from each cluster, top k sentences are selected for the formation of the final multi-document summary. The sentences in the final summary maintain the same order of their position as in the source documents.

**Sentence Clustering**: Sentence similarity is used to perform clustering of single document summaries. The sentence similarity for clustering is calculated using syntactic and semantic similarity measures proposed by Liu [10].

**Syntactic Similarity**: Liu et al. [10] used a method to calculate the syntactic similarity between two sentences using their word order. Each word is assigned a unique index which is used to represent an original order ($v_0$) and a relative order ($v_r$). The index number of the first sentence represents the original order. Common words in both the sentences are used to create relative order vector.

For example, the original and the relative word order vector for the two sentences *The building is taller than the pole* (S1) and *The pole is taller than the building* (S2) is calculated as below:

Index no. for $S_1$: {1, 2, 3, 4, 5, 6, 7}

Index no. for $S_2$: {1, 2, 3, 4, 5, 6, 7}

Original order vector $v_0$= {1, 2, 3, 4, 5, 6, 7}

Relative order Vector $v_r$= {1, 7, 3, 4, 5, 6, 2}

Liu et al. [10] used correlation coefficient between the original and relative vector to calculate the similarity:

$$Sim_0(S_1, S_2) = \frac{\sum(v_0 * v_r) - \frac{\sum v_0 * \sum v_r}{k}}{\sqrt{(\sum {v_0}^2 - \frac{(\sum v_0)^2}{k})(\sum {v_r}^2 - \frac{(\sum v_r)^2}{k})}}$$

Where k represents the total number of words in an original sentence. Syntactic similarity can have maximum value = 1, when the S1 and S2 word order is identical.

**Semantic similarity:** Li et al. [8] proposed a method for computing the semantic similarity. First, WordNet [40] is used to calculate the semantic similarity between words. Semantic similarity between sentences is obtained by adding these calculated word similarities. Words are arranged in a Semantic based hierarchy by WordNet.

**Semantic similarity between words**: Semantic similarity between words is computed by using an edge count based method. If the words have common features more than different features, they are assumed to be more similar. Common and different features between words are determined by the path length and depth of subsume in Wordnet hierarchy.

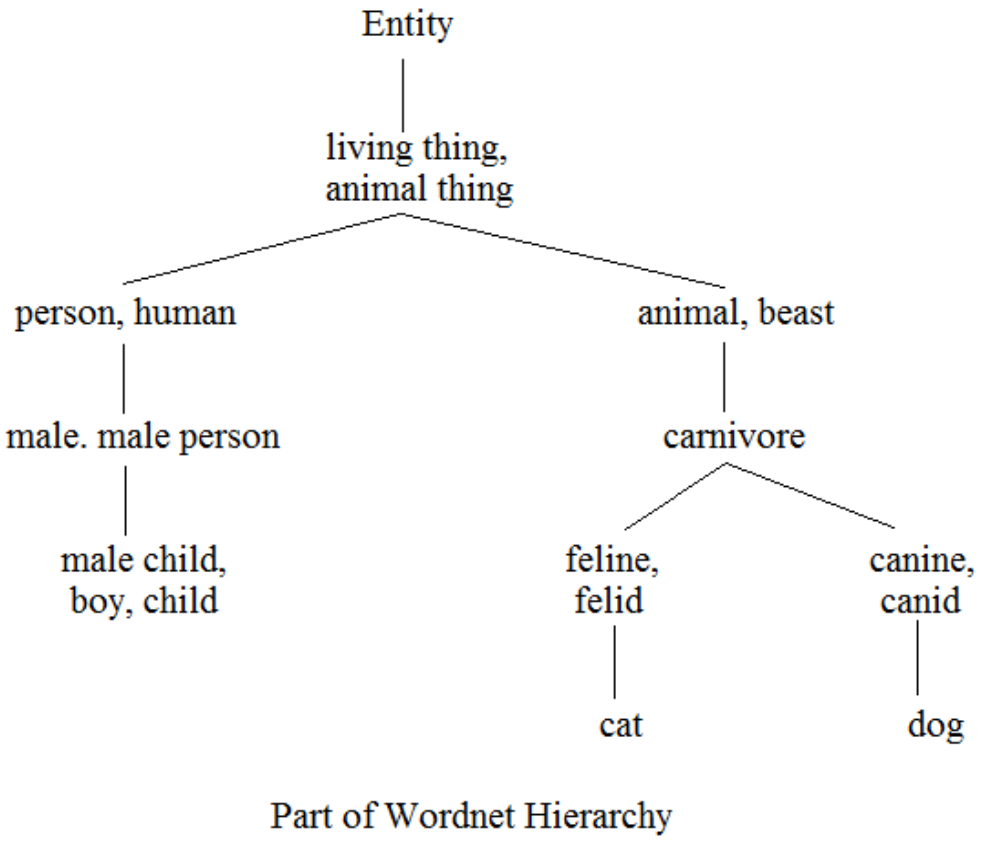


**Figure 4.5: Parts of Word net Hierarchy**

- **Shortest Path Length (*l*):** It is the shortest path distance between two words in Wordnet hierarchy. Lesser the length of the shortest path between two words, more similar they are and vice-versa. For the same words, the shortest path length is 0.
- **Depth of Subsumer**: The depth of subsumer (*d*) is described as the length of the common word between two words [8, 10]. The more is the value of depth, the less will be the similarity between the words. The semantic similarity can be calculated as :

$$S_w(w_1, w_2) = \frac{f(d)}{f(d)+f(l)}$$

Where d represents the depth of subsumer, l represents shortest path length and f is a transfer function i.e. $f(x) = e^x$-1.

Semantic similarity value may vary between 0 and 1. If the two words are exactly similar then 1 and 0 for dissimilar words.

When d= 0, no common parent, then, $S_w(w_1, w_2) = 0$;

When l = 0, same synset, and $S_w(w_1, w_2) = 1$.

If both d and l are non-zero then the similarity can be calculated as:

$$S_w(w_1, w_2) = \frac{e^{\alpha d}-1}{e^{\alpha d}+e^{\beta l}-2} \quad (0 < \alpha, \beta \leq 1)$$

Where α and β represent smoothing factors.

- **Information Content**: It is a measure of information represented by a word and is calculated as:

$$I(w) = -\frac{\log p(w)}{\log(N+1)}$$

For calculating the frequency of words British National Corpus [12] is used. The corpus is huge and contains more than 100 million words. The probability of words is computed as:

$$P(w) = \frac{n+1}{N+1}$$

Where *n* represents the frequency of the word in the corpus and *N* is the total number of words in the corpus.

Finally, the Semantic similarity is calculated by information content and semantic similarity between words, calculated as follows:

$$Sim_s(S_1, S_2) = \frac{\sum_{w_i \in S_1} \max_{w_j \in S_2} \left(S_w(w_i, w_j) * I_{w_i}\right)}{\sum_{w_i \in S_1} I_{w_i} + \sum_{w_j \in S_2} I_{w_j}}$$

Where *I(w)* represents the information content and $S_w(w_1, w_2)$ is the semantic similarity between words. The overall similarity between two sentences is computed as:

$$Sim_{sen} = Sim_s(S_1, S_2) * \left((1-\gamma) + \gamma * Sim_0(S_1, S_2)\right) + Sim_s(S_2, S_1) * \left((1-\gamma) + \gamma * Sim_0(S_2, S_1)\right)$$

Where $\gamma$ represents smoothing factor.

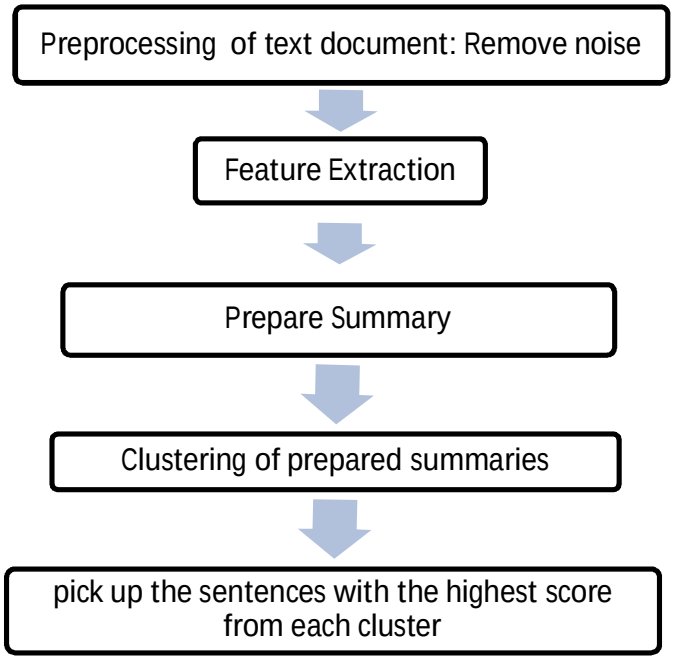


**Figure 4.6: Flowchart for Sentence Clustering Algorithm**

**4. Multi Document Summary:** The sentences of single document summaries are clustered using sentence similarity. From each cluster, top k sentences are selected to form final summary. The extracted sentences are sorted according to their actual position in the original source document to prepare the multi-document summary.

# Chapter 5

# Comparison and Evaluation

This chapter defines the measures: Similarity Score, ROUGE[12] and BLEU [13]metric, used to check the quality and accuracy of the system generated summaries. The accuracy of the single document and Multi-document Summarization algorithms, described in the previous chapters, is evaluated on the basis of these measures.

## 5.1 Measures

We have examined the summary of all the explained datasets by the previously described algorithms. Then we have evaluated the accuracy of each algorithm generated summary against the set of Human prepared summaries. The human prepared summary is assumed to have the highest accuracy as it includes the human understanding and evaluation. We have checked for the similar words in both the summaries and the provided a similarity score for each one of the algorithms produced summary.

**Similarity Score** is a measure used for checking similarity among text data. It considers the common words and the position of words between system generated summary and human prepared summary. It returns the similarity score value in the range of 0 to 1.

**ROUGE** (Recall-Oriented Understudy for Gisting Evaluation) measure is the most popular measure to identify the quality of system generated summary. It is a content overlap measure which tests how an algorithm generated summary matches to the reference summaries produced by human interpretation and understanding. It is a recall-based measure which encourages the algorithms and systems to consider all the key topics in the summary. Recall measure can be calculated by using unigrams, bigrams or trigrams matching. For example, ROUGE-1 is evaluated as a count of unigrams in the system generated summary and reference summary.

If there are multiple summary references, the evaluated ROUGE-1 scores are averaged. ROUGE can only find out if the similar key concepts are discussed between system summary and human generated summary, but the coherence of the sentences cannot be checked. High-order n-gram ROUGE measures try to determine fluency of the summary.

**BLEU metric**

BLEU metric can be described as a modified form of precision, generally used for machine translation evaluation.

Precision represents the ratio of the number of common words in both gold and model translation/summary to that present in the model summary. Unlike ROUGE, BLEU takes the weighted average and directly accounts for variable length phrases.

The actual metric is just precision which is modified to avoid the problem when a model's translation/summary contains redundant information.

## 5.2 Single Document Summarization Algorithms

This subsection contains the comparison tables with evaluated scores for Single document summarization algorithms for the datasets described in APPENDIX 1.

- News blog – Demonetisation Dataset
- Medical domain - Alzheimer's Dataset
- Cricket related Dataset

Table 5.1, 5.2 and 5.3 represent the similarity score of summaries for different datasets.

Table 5.1 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for News blog - Demonetisation dataset .

**Table 5.1: Similarity score for News blog- Demonetisation Dataset**

| Algorithm | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **TextRank** | 0.72 | 0.576 | 0.269 |
| **Texteaser** | 0.58 | 0.473 | 0.311 |
| **Smmry tool** | 0.52 | 0.475 | 0.206 |

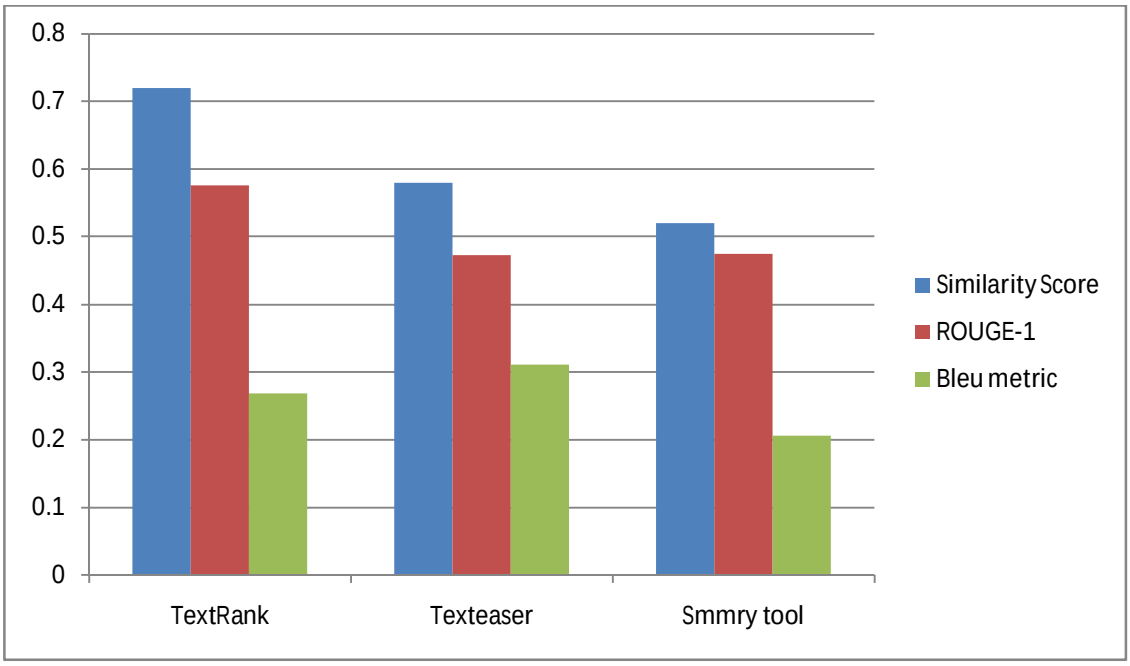


**Figure 5.1: Graphical representation-News blog- Demonetisation Dataset**

Table 5.2 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for Medical- Alzheimer's dataset .

**Table 5.2: Similarity score for Medical- Alzheimer's Dataset**

| Algorithm | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **TextRank** | 0.298 | 0.263 | 0.197 |
| **Texteaser** | 0.411 | 0.357 | 0.251 |
| **Smmry tool** | 0.493 | 0.417 | 0.323 |

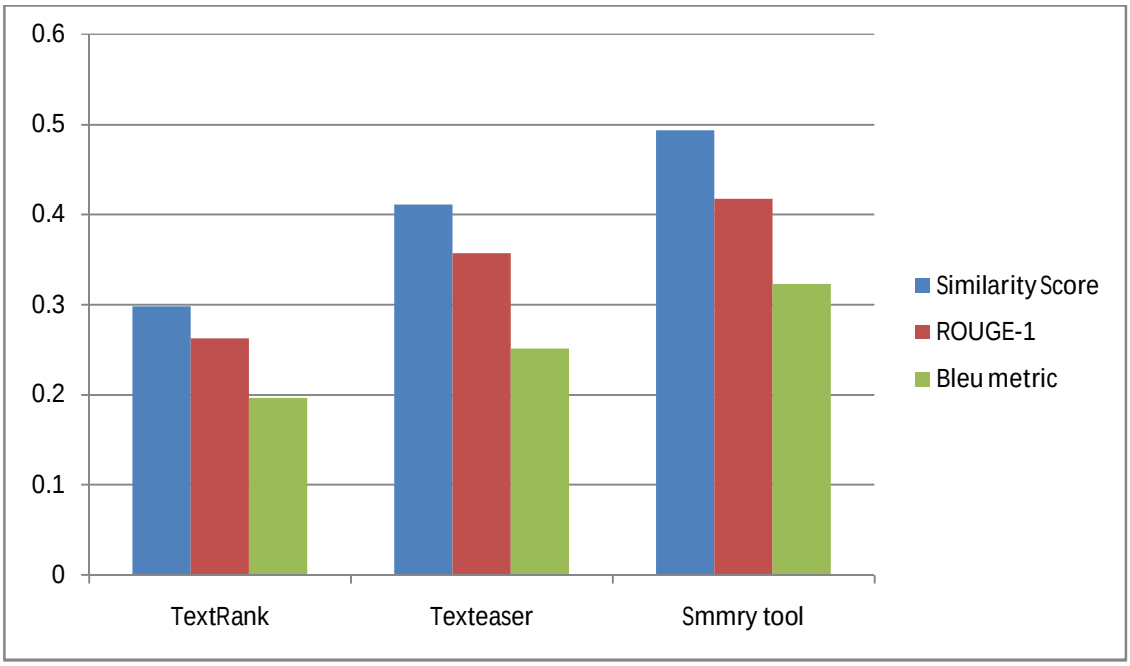


**Figure 5.2: Graphical representation- Medical- Alzeihmer's Dataset**

Table 5.3 and Figure 5.3 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: TextRank, Texteaser and Summary by word features for Cricket related dataset .

**Table 5.3: Similarity score for Cricket related Dataset**

| Algorithm | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **TextRank** | 0.51 | 0.436 | 0.255 |
| **Texteaser** | 0.39 | 0.298 | 0.134 |
| **Smmry tool** | 0.39 | 0.264 | 0.211 |

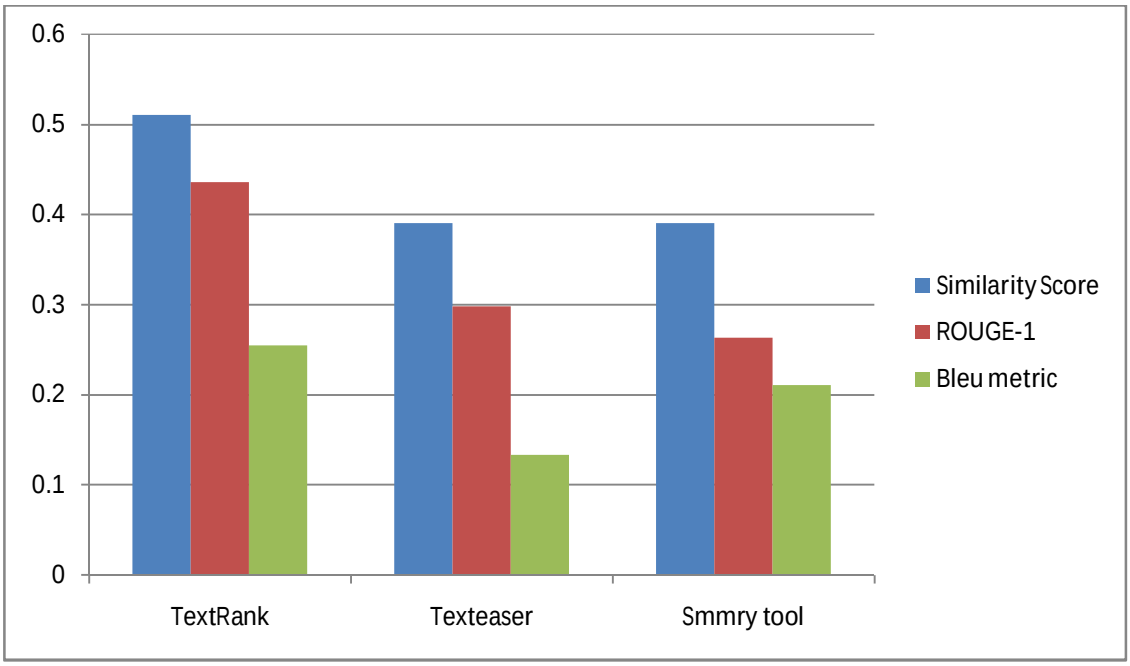


**Figure 5.3: Graphical representation- Cricket related Dataset**

From these tables, we have analyzed that the cricket domain data is best summarized by the TextRank Algorithm.

For medical dataset, Summary based on word features gives similarity score of 0.51, best among all other algorithms.

For News related dataset, text Rank gives 0.72 similarity score.

## 5.2 Multi-document Summarization Algorithms

The tables 5.4, 5.5 and 5.6 represent the similarity score of summaries for different datasets.

Table 5.4 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: Multi-document Summarization using ILP based method, LDA topic model and Summarization based on sentence clustering for News blog- Demonetisation Dataset.

**Table 5.4: Similarity score for News blog- Demonetisation Dataset**

| Algorithm based on | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **ILP based Sentence Fusion** | 0.24 | 0.29 | 0.18 |
| **LDA topic Model** | 0.38 | 0.431 | 0.34 |
| **Sentence Clustering** | 0.402 | 0.41 | 0.14 |

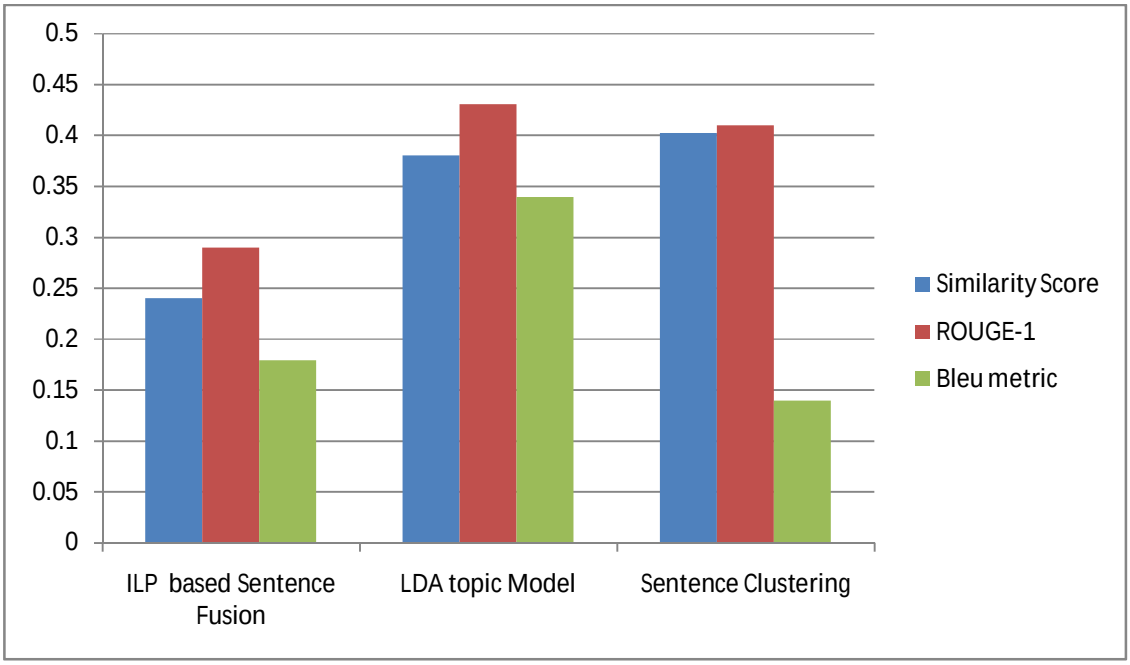


**Figure 5.4: Graphical Representation-News blog- Demonetisation Dataset**

Table 5.5 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: Multi-document Summarization using ILP based method, LDA topic model and Summarization based on sentence clustering for Medical- Alzheimer's Dataset.

**Table 5.5: Similarity Score for Medical- Alzheimer's Dataset**

| Algorithm based on | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **ILP based Sentence Fusion** | 0.21 | 0.23 | 0.17 |
| **LDA topic Model** | 0.27 | 0.35 | 0.19 |
| **Sentence Clustering** | 0.43 | 0.34 | 0.28 |

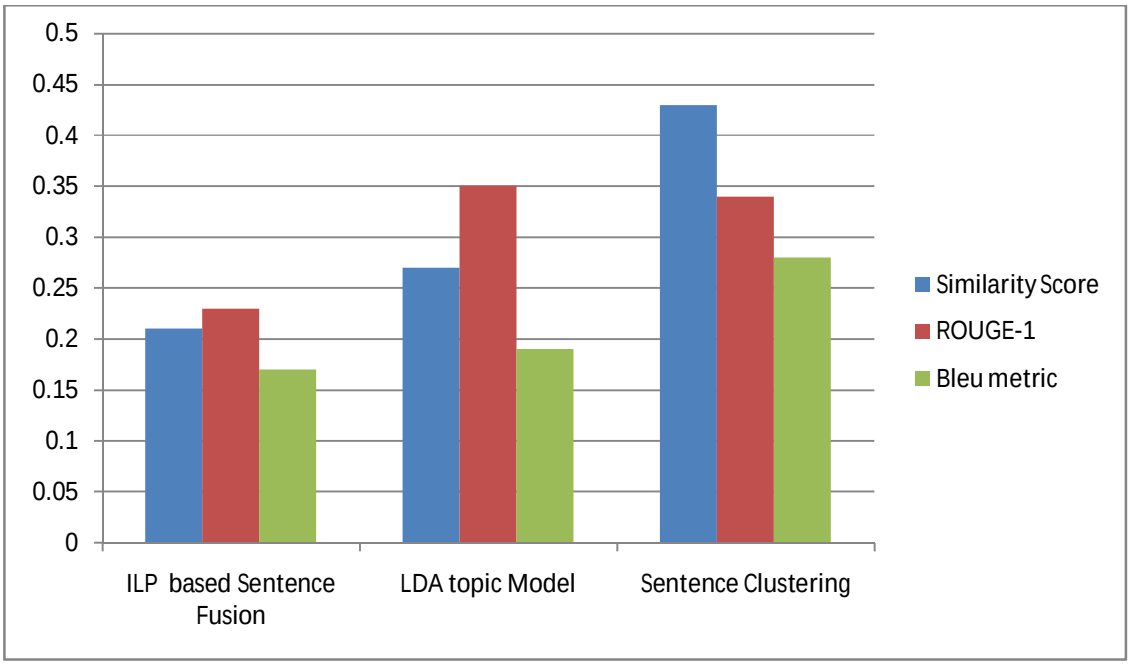


**Figure 5.5: Graphical Representation- Medical- Alzeihmer's Dataset**

Table 5.6 describes the evaluated scores calculated by the Single document summaries prepared by algorithms: Multi-document Summarization using ILP based method, LDA topic model and Summarization based on sentence clustering for Cricket related Dataset.

**Table 5.6: Similarity Score for Cricket related Dataset**

| Algorithm based on | Similarity Score | ROUGE-1 | Bleu metric |
|---|---|---|---|
| **ILP based Sentence Fusion** | 0.18 | 0.21 | 0.11 |
| **LDA topic Model** | 0.29 | 0.31 | 0.24 |
| **Sentence Clustering** | 0.36 | 0.35 | 0.31 |

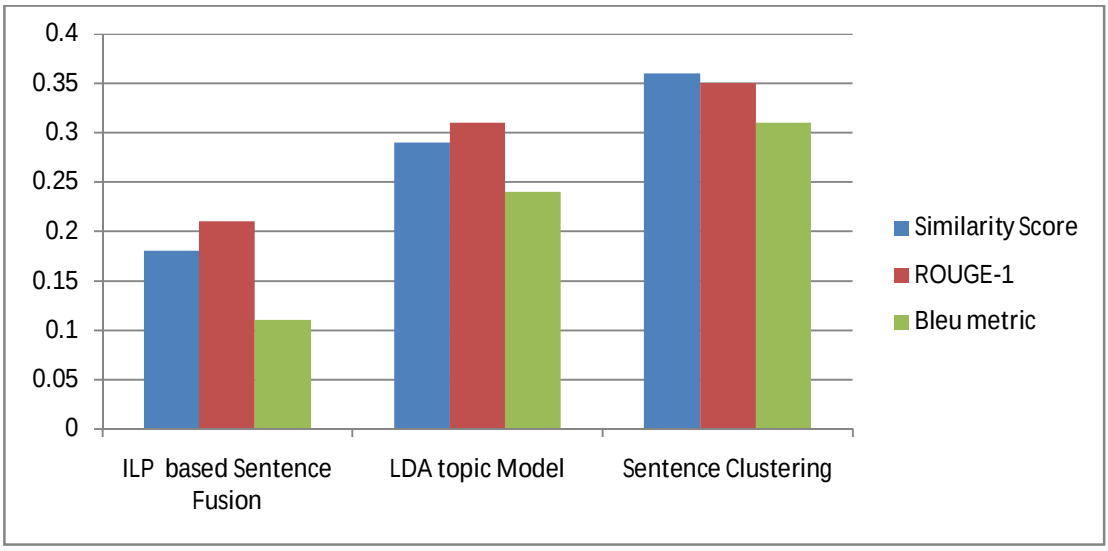


**Figure 5.6: Graphical Representation- Cricket Related Dataset**

From the above three tables, we have analyzed that the News domain- Demonetisation data is best summarized by the Multi-document Summarization based on LDA Topic Model.

For medical dataset and cricket dataset, Multi-document Summarization based on Sentence Clustering outperforms the other two algorithms.

Sentence Clustering based algorithm gives better results because of its sentence clustering of single document summaries and extractive nature.

# Chapter 6

# Conclusions and Future Scope

Automatic Text Summarization is used to get an important piece of text from a larger document. A large number of algorithms designed and implemented to get a good, coherent and non-redundant summary a little similar to the human prepared summary.

Simple single document extractive algorithms have given better results in different domains as compared to abstractive summarization algorithms.

Extractive summarizers are used to select the important set of sentences from the source document based on top scoring Sentence-ranking method. These methods use different feature extraction and content selection methods like upper case words, the frequency of words, similarity chains, logical closeness etc. for selecting summary sentences.

Abstractive Summarizers prepare new sentences based on the important information existing in the source document. These summarizers make new sentences by the union of multiple sentences. They use word graphs to select a set of words to produce a coherent sentence.

Based on the Comparison results, it can be seen that by performing Automatic Text Summarization to get a gist of the input text documents equivalent to human interpreted summary is not yet fulfilled, but by improving the existing algorithms, the value of evaluation metrics is increasing.

## Future Scope

With the rapid increase in the electronic data on internet and less time to read the documents based on a similar topic has called a need to design accurate and efficient Multi- document summarization systems. As research on text summarization started 50 years ago and a lot of work has been done in the extractive area in both the single and multiple document domains but there is still a long path to cover in this field. Abstractive summarizers aim to import more information in a single sentence rather than include the sentence as a whole.

Multi-document Abstractive Summarization is the area which is needed to be explored.

Over time, attention has drifted from summarizing scientific articles to news articles, electronic mail messages, advertisements, and blogs. Domain associated summarizes can be a solution to get more accurate summaries. Medical and Legal matters domain can be highly benefitted from this area of research even if they focus only on small details related to a general summarization process and not on building an entire domain dependent summarization system.

# Appendix-I

# Datasets

In this Chapter, The input datasets are defined:

**Dataset 1:**

Dataset 1 includes the text files containing the data from newspapers, internet and news blogs regarding the news demonetization. It involves the date on which notes of 500 and 1000 were declared illegal tender. It includes the date on which it was announced. Various rules imposed time to time and expert views on the move. Problems faced by the citizens and rules formed to facilitate the people are also specified. Total collection and immediate effects are also stated in the articles.

| | |
|---|---|
| Number of characters (including spaces) : | 40362 |
| Number of characters (without spaces) : | 32625 |
| Number of words : | 6728 |
| Lexical Density : | 27.1106 |
| Number of sentences : | 417 |
| Number of syllables : | 11214 |

**Output Summary**

Here, we expect the summary to include how it started, effects and final results and reason associated with the news.

## Dataset 2:

Dataset 2 is a medical related data about a disease called Alzheimer's. Alzheimer's is a disease which causes the patient to forget about the things and then it proceeds to a point where patient may no longer recognize their family members or the thing which has been done even before 5 minutes. The dataset includes the definition and introduction to the problem. Then it analyses the causes associated which are likely to cause Alzheimer's. It also includes the cure and how to approach the disease in the first phase. Different phases are described with the help of conditions of a patient.

| | |
|---|---|
| Number of characters (including spaces) : | 37045 |
| Number of characters (without spaces) : | 29515 |
| Number of words : | 5886 |
| Lexical Density : | 23.1057 |
| Number of sentences : | 356 |
| Number of syllables : | 10079 |

**Output Summary**

For this dataset, the summary is expected to include a brief introduction to the disease and then a little information about the different phases and cure for the problem.

## Dataset 3:

This dataset includes cricket related data. It includes the history of cricket in India how it started and various milestones achieved in the times. About The Indian team lifted the World Cup and a little about the prominent players. It also includes the cricket control bodies on Indian as well as the international level i. e. the ICC and BCCI, and about how they work and organize the international tournaments regularly. Our current team captains and teammates related data are also included.

| | |
|---|---|
| Number of characters (including spaces) : | 29302 |
| Number of characters (without spaces) : | 23494 |
| Number of words : | 5046 |
| Lexical Density : | 23.8605 |
| Number of sentences : | 234 |
| Number of syllables : | 8129 |

**Output Summary**

Here for this dataset we expect the summary to include a little history of the cricket and years when the Indian team won the international tournaments. A little about the control bodies and current cricket team and coach shall also be included in the summary.

# REFERENCES


[1] D.K. Gaikwad and C.N. Mahender "A Review Paper on Text Summarization", International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 3, March 2016,154-160

[2] Radev, D. R., Hovy, E., and McKeown, K. (2002) "Introduction to the special issue on summarization." Computational Linguistics., 28(4):399-408

[3] Luhn, H. P. (1958) " The automatic creation of literature abstracts". IBM Journal of Research Development, 2(2):159–165

[4] Edmundson, H. P. (1969) " New methods in automatic extracting". Journal of the ACM, 16(2):264–285.

[5] R.Mihalcea, and P.Tarau, "TextRank: Bringing Order into Texts." In Proceedingsof Empirical Methods in Natural Language Processing (EMNLP). pp. 404-411. 2004.

[6] Z.Pei-ying, and L.Cun-he, "Automatic Text Summarization based on Sentences Clustering and Extraction," Proceeding of the 2nd IEEE International Conference on Computer Science and Information Technology. pp. 167-170. 2009

[7] 20IOy International Conference on Computer Application and System Modeling (ICCASM 2010) Automatic Text Summarization Based On Rhetorical Structure Theory Li Chengcheng 595-598

[8] D. Blei, A. Ng, and M. Jordan " Latent Dirichlet allocation". In Journal of Machine Learning Research, 3:993–1022, January2003.

[9] Barzilay, R. and Elhadad, M. (1997). "Using lexical chains for text summarization." in Proceedings ISTS'97. pg. 38-41

[10] Radev, D. R. and McKeown, K. (1998) "Generating natural language summaries from multiple on-line sources." Computational Linguistics, 24(3):469–500

[11] S. Banerjee, P.Mitra and K. Sugiyama " Multi-Document Abstractive Summarization Using ILP Based Multi-Sentence Compression"in Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

[12] Lin, C.-Y. (2004). "Rouge: A package for automatic evaluation of summaries." In Proceedings of the ACL-04 Workshop, pages 74–81, Barcelona, Spain

[13] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu “BLEU: a Method for Automatic Evaluation of Machine Translation” in Computational Linguistics (ACL), Philadelphia, July 2002, pp. 311-318

[14] S. Brin and L. Page “The PageRank Citation Ranking:Bringing Order to the Web” in 1999

[15] Mc Keown, K. R. and Radev, D. R. (1995). “Generating summaries of multiple news articles.” in Proceedings of SIGIR ’95, pages 74–82, Seattle, Washington.

[16] Jagadeesh J, Prasad Pingali, Vasudeva Varma “Sentence Extraction Based Single Document Summarization” Workshop on Document Summarization, 19th and 20th March, 2005, IIIT Allahabad

[17] Kamal Sarkar, “Sentence Clustering-based Summarization of Multiple Text Documents”, TECHNIA – International Journal of Computing Science and Communication Technologies, vol. 2, no. 1, Jul. 2009.

[18] F. Canan Pembe and Tunga Güngör, “Automated Query-biased and Structure-preserving Text Summarization on Web Documents,” in Proceedings of the International Symposium on Innovations in Intelligent Systems and Applications, İstanbul, June 2007.

[19] Reeve Lawrence H., Han Hyoil, Nagori Saya V., Yang Jonathan C., Schwimmer Tamara A., Brooks Ari D., “Concept Frequency Distribution in Biomedical Text Summarization”, ACM 15th Conference on Information and Knowledge Management (CIKM), Arlington, VA, USA,2006.

[20] Khan Atif, Salim Naomie, “A review on abstractive summarization Methods”, Journal of Theoretical and Applied Information Technology, 2014, Vol. 59

[21] Evans, D. K. (2005). “Similarity-based multilingual multi-document summarization.” Technical Report CUCS-014-05, Columbia University.

[22] Edmundson, H. P. (1969). “New methods in automatic extracting.” Journal of the ACM, 16(2):264–285.

[23] Martins, Camilla Brandel and Lucia Helena Machado Rino. “Revisiting UNLSumm: Improvement Through a Case Study.” (2002).

[24] Conroy, J. M. and O’leary, D. P. (2001). “Text summarization via hidden markov models.” In Proceedings of SIGIR ’01, pages 406–407, New York, NY, USA

[25] Kupiec, J., Pedersen, J., and Chen, F. (1995). “A trainable document summarizer.” In Proceedings SIGIR ’95, pages 68–73, New York, NY, USA.

[26] Aone, C., Okurowski, M. E., Gorlinsky, J., and Larsen, B. (1999). “A trainable summarizer with knowledge acquired from robust nlp techniques”.pages 71–80

[27] Lin, C.-Y. and Hovy, E. (1997). “Identifying topics by position.” In Proceedings of the Fifth conference on Applied natural language processing, pages 283–290, San Francisco, CA, USA.

[28] Osborne, M. (2002). Using maximum entropy for sentence extraction. In Proceedings of the ACL’02 Workshop on Automatic Summarization, pages 1–8, Morristown, NJ, USA

[29] Svore, K., Vanderwende, L., and Burges, C. (2007). “Enhancing single-document summarization by combining RankNet and third-party sources.” In Proceedings of the EMNLP-CoNLL, pages 448–457.

[30] Barzilay, R. and Elhadad, M. (1997). “Using lexical chains for text summarization.” in Proceedings ISTS’97.

[31] Hovy, E. and Lin, C. Y. (1999). “Automated text summarization in summarist.” In Mani, I. and Maybury, M. T., editors, Advances in Automatic Text Summarization, pages 81–94. MIT Press

[32] N. Aletras and M. Stevenson. “Evaluating topic coherence using distributional semantics.” In Proc. Of the 10th Int. Conf. on Computational Semantics (IWCS’13), pages 13–22, 2013.

[33] Kamal Sarkar “Automatic Single Document Text Summarization Using Key Concepts in Documents” J Inf Process Syst, Vol.9, No.4, pp.602-620, December 2013

[34] I. Chen “Integer Linear Programming Models for Constrained Clustering” in International Conference on Discovery Science 2010: Discovery Science pp 159-173

[35] Günes Erkan and Dragomir R. Radev. 2004. “LexRank: graph-based lexical centrality as salience in text summarization”. J. Artif. Int. Res. 22, 1 (December 2004), 457-479.

[36] Jinqiang Bian, Zengru Jiang, Qian Chen 2014 “Research On Multi-document Summarization Based On LDA Topic Model” Sixth International Conference on Intelligent Human-Machine Systems and Cybernetics 113-116

[37] Virendra Kumar Gupta Tanveer J. Siddiqui “Multi-Document Summarization Using Sentence Clustering” IEEE Proceedings of 4th International Conference on Intelligent Human Computer Interaction, Kharagpur, India, December 27-29, 2012

[38] The Porter Stemming Algorithm [Online] Available:http://tartarus.org/~martin/PorterStemmer/

[39] George A. Miller. “WordNet: A Lexical Database for English.” Communications of the ACM, pages 39-41, November 1995

[40] Sherry and Dr. P. Bhatia “ A Survey to Automatic Text Summarization Techniques” International Journal of Engineering Reasearch, October 2015 Pg. 1045- 1053

[41] Chin-Yew Lin and Eduard Hovy, “Identifying Topics by Position,” In Proceedings of the Fifth conference on Applied natural language processing, San Francisco, pp. 283-290, 1997.

[42] S. P. Yong, A. I. Z. Abidin and Y. Y. Chen, “A Neural Based Text Summarization System,” 6th International Conference of Data Mining, pp. 45-50, 2005.

[43] Ruqaiya Hasan, Coherence and Cohesive Harmony, In: Flood James (Ed.), Understanding Reading Comprehension: Cognition, Language and the Structure of Prose. Newark, Delaware: International Reading Association, pp. 181-219, 1984.

[44] William C. Mann and Sandra A. Thompson, Relational Propositions in Discourse, Defense Technical Information Center,

[45] Branimir Boguraev and Christopher Kennedy, “Saliencebased Content Characterization of Text Documents,” In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, 1997.

[46] Li Chengcheng, “Automatic Text Summarization Based On Rhetorical Structure Theory,” International Conference on Computer Application and System Modeling (ICCASM), vol. 13, pp. 595-598, October 2010.

[47] Xiaojun Wan, “An Exploration of Document Impact on Graph-Based Multi-Document Summarization,” Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics,

[48] Tiedan Zhu and Xinxin Zhao, “An Improved Approach to Sentence Ordering For Multi-document Summarization,” IACSIT Hong Kong Conferences, IACSIT Press, Singapore, vol. 25, pp. 29-33, 2012.