

Graph and Rhetoric Structure based Learning Framework for Extractive Multi-Document Summarization



Ph.D. (Computer Science) Thesis

By

Zakia Jalil

Registration No: 102-FBAS/PHDCS/F14

Supervisor: Dr. Tehmina Amjad (IIUI, Pakistan)

Co-Supervisor: Dr. Jamal Abdul Nasir (NUI Galway, Ireland)

Department of Computer Science, Faculty of Computing & Information
Technology, International Islamic University, Islamabad, Pakistan

May 2023



Accession No. TH-27408 K

PHD

006.312

ZAC

Computer science

Summarization

Automatic summarizing (computer)

Machine learning

Artificial intelligence

**INTERNATIONAL ISLAMIC UNIVERSITY ISLAMABAD
FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE**

Date: 26-05-2023

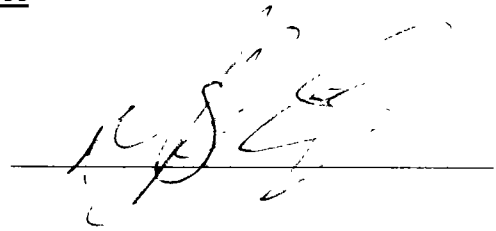
Final Approval

It is certified that we have read the thesis titled "Graph and Rhetoric Structure based Learning framework for Extractive Multi-Document Summarization" submitted by Ms. Zakia Jalil (102-FBAS/PHDCS/F14). It is our conclusion that this thesis is of sufficient standards to warrant its acceptance by the International Islamic University, Islamabad for the PhD Degree in Computer Science.

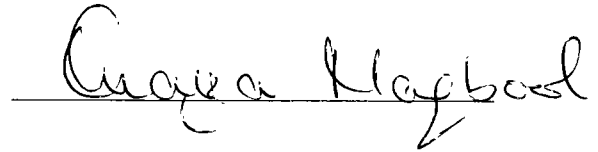
Committee

External Examiners:

Dr. Muhammad Shiraz
Associate Professor,
Department of Computer Science
FUUAST, Islamabad



Dr. Onaiza Maqbool
Professor,
Department of Computer Science
Quaid e Azam University, Islamabad



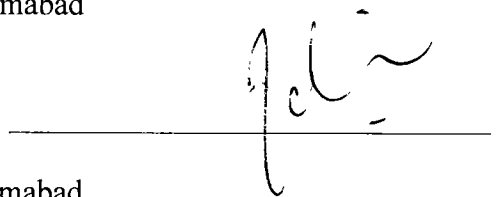
Internal Examiner:

Dr. Umara Zahid
Lecturer,
Department of Computer Science, FC&IT, IIU, Islamabad



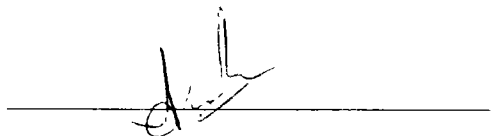
Supervisor:

Dr. Tehmina Amjad
Assistant Professor,
Department of Computer Science, FC&IT, IIU, Islamabad



Co-Supervisor:

Dr. Jamal Abdul Nasir,
Assistant Professor,
School of Computer Science, University of Galway, Ireland



This dissertation is submitted to International Islamic University Islamabad,
Pakistan in partial fulfilment of the requirement of the degree of Doctor of
Philosophy (Computer Science)

Declaration

I earnestly declare that this thesis neither as a whole nor as part, has been copied out from any source. It is further declared that I have completed this thesis entirely on the basis of my personal effort, made under the sincere guidance of my supervisors. I also declare that the work presented in this report has not been submitted in support of any other application or degree or qualification in any other University or Institute.

Zakia Jalil

102-FBAS/PHDCS/F14

Dedication

This work is dedicated to my beloved parents whose constant support enable me to achieve
this milestone,

To my husband who never let me give up during the most difficult times encountered,

To my siblings, who are the most understanding people I have ever seen,

And to my colleagues and friends, who believed in me while situations were not friendly.

Zakia Jalil

102-FBAS/PHDCS/F14

Acknowledgements

I pay my submissive gratitude for Almighty Allah who has bestowed me the strength and qualities due to which I have been able to complete my research.

I would like to acknowledge the support and contributions of my Supervisor Dr Tehmina Amjad who has always been the main intuition and inspiration of my hard work. I am thankful for her valuable ideas, suggestions and experience that motivated me to complete the research thesis. Special thanks to my supervisor Dr Jamal Abdul Nasir, NUI Galway, Ireland, for his valuable suggestion and innovative ideas to complete my PhD research.

I owe my loving thanks to my parents Mr Noor Jalil and Mrs Anwar Sultana(late), my loving husband Mr Muhammad Nasir, my beloved sister Nazia, my siblings, and Manahil, for sacrifice of time and provision of peaceful environment conducive for studies and accomplishment of work.

I am also thankful to my colleagues at International Islamic University for their cooperation and support to accomplish this task. I am especially thankful to Dr Sadia Arshid (late) for inspiring me to complete my PhD, and then Ms Shaista Rashid to thoroughly listening to my problems for hopeful suggestions. I am thankful to my friend Ms Maria Ashraf who helped me more like friend than a colleague. Special thanks to Ms Sabina Irum and Ms Azka Atiq who motivated me more when I was ready to give up. I would like to thank Dr Umara Zahid and her husband for their valuable suggestions all through my thesis.

Zakia Jalil

102-FBAS/PHDCS/F14

Abstract

Context: With the tremendous growth in the number of electronic documents, it is challenging to manage the volume of information. A significant amount of research has been focused on automatic summarization of the information available in the text documents. Multi-Document Summarization (MDS) is one approach that aims to extract the information from the available documents.

Research Gap: The process of extractive summarization renders compromised grammaticality since it extracts the important words or sentences from the given documents to prepare the desired summary.

Proposed Solution: This research endeavour aims to explore and exploit extractive MDS using learning-based Graph Theory and Rhetoric Structure Theory to address the challenges likes grammaticality, coverage, informativity, diversity, and redundancy mitigation.

Method: In this regard, we chose multi-method research methodology. We first conducted an extensive literature survey to identify the research progress focusing on MDS. Later, we designed and presented two novel extractive MDS systems namely Grapharizer and RSTSummarizer and tested it on two benchmark datasets, i.e., DUC 2004 and recent news articles. We further test the role of machine learning (ML) in generating better summaries.

Results: The proposed methods improve the grammaticality, informativity, and diversity of the produced summaries and reduced the redundancy as well. We test our methods against the state-of-the-art methods in the same category as well as across domain like term based, ontology based, closed pattern-based, and ChatGPT methods. The results of proposed methods are promising. The results show significant improvements with ML as well. The evaluations using ROUGE 2.0 variants depict comparable results by Grapharizer and Grapharizer with ML, and RSTSummarizer when compared with the various baselines techniques. The user evaluation of summaries was also performed, and they ranked the accuracy of synonym mapping with 88%, informativity of summary with 84%, and redundancy mitigation with 88% in human evaluation phase.

Conclusion: This thesis establishes that extractive Multi-Document Summarization using Graph Theory and Rhetoric Structure Theory with its learning-based variants gives promising results to address the challenges likes grammaticality, informativity, diversity, and redundancy mitigation on DUC 2004 and recent news articles. Moreover, we conclude that pre-processing the dataset with synonym mapping, multi-word expressions, pronoun replacement, and topic modelling significantly improves the quality of the summary. **Keywords:** Extractive Multi-Document Summarization, Topic Modeling, ChatGPT, Machine Learning, Word-Graph, RST.

Published Papers:

1. Extractive Multi-Document Summarization: Review of progress in the last decade- IEEE Access. (**Impact factor = 3.476**)
2. “What have you read” based multi-document summarization- International Journal of Innovations in Science & Technology, 4(5), 94-102.
3. Grapharizer: Graph based multi-document summarization- Big Data Analytics and Artificial Intelligence in Electronics - Special Issue (**Impact factor = 2.69**)

Under review:

RST summarizer for Extractive Multi-document Summarization

Un-submitted paper:

Extractive Multi-Document Summarization Framework: ChatGPT versus Champions

Table of Contents

1	Introduction	1
1.1	Single Document Summarization.....	1
1.2	Multi Document Summarization.....	2
	i) Abstractive Summarization.....	2
	ii) Extractive Summarization.....	3
1.3	Research Motivation	4
1.4	Thesis Contribution.....	5
1.5	Thesis organization	6
2	Related Work.....	8
2.1	Literature Review	9
2.1.1	Ontology-Based Methods	10
2.1.2	Term-Based Methods.....	11
	2.1.2.1 Clustering-Based Methods	13
	2.1.2.2 Latent Semantic Analysis (LSA) Methods.....	19
	2.1.2.3 Non-Negative Matrix Factorization (NMF) Methods.....	25
2.1.3	Rhetoric Structure Theory-Based (RST) Methods	25
2.1.4	Graph-Based Methods	26
2.1.5	Miscellaneous Methods	32
2.1.6	Secondary Studies Conducted in MDS.....	34
2.2	Datasets	35
2.3	Evaluation Techniques	35
2.4	Research Gap and Limitations	36
3	Research Method	39
3.1	Research Objectives and Contributions	39
3.2	Research Questions	40
3.3	Proposed Solution	41
	3.3.1 Module 1	41
	3.3.2 Module 2- The RST based Extractive MDS.....	46
	3.3.3 Module 3: Machine Learning for EMDS.....	47
	3.3.4 Module 4- Baselines comparison.....	49
3.4	Research Methodology.....	50
	3.4.1 Definition.....	50
	i) Representativeness.....	51

ii) Diversity.....	52
iii) Length of the desired summary.....	52
iv) Redundancy mitigation	52
v) Grammaticality	53
3.4.2 Planning	53
3.4.3 Baselines	54
3.4.4 Dataset: DUC-Document Understanding Conference:.....	55
4 Graph Model.....	56
4.1 Background	58
4.2 The proposed technique	62
4.2.1 Motivation.....	62
4.2.2 Grapharizer: The Graph-Based Method.....	63
4.2.2.1 Pre-processing	64
4.2.2.2 Overview of The Graph Generation Process.....	66
4.2.2.3 Representativeness	68
4.2.2.4 Removing The Redundancy	68
4.2.2.5 Grammaticality.....	69
4.3 Experiment and Evaluation	70
4.3.1 Dataset.....	70
4.3.2 Evaluation Metric.....	71
4.3.3 Evaluation and Comparison.....	71
4.4 Conclusion.....	78
5 RST Model	80
5.1 The proposed technique	81
6 Implementations of Baselines.....	84
6.1 Term-based method.....	84
6.2 Ontology-based method	84
6.3 Close patterns-based method.....	85
6.3.1 Generating Closed Patterns.....	86
6.3.2 Sentence-Representation.....	86
6.3.3 Sentence-Ranking	86
6.3.4 Sentence-Selection.....	86
6.4 Experiment and discussion.....	87
7 Machine Learning.....	92
7.1 Support Vector Machines.....	92

7.2	Artificial Neural Networks.....	92
7.3	Multivariate Linear Regression.....	92
8	Results and Discussion	100
8.1	Comparison of Results with related work.....	100
8.2	Quantitative Analysis	100
8.3	Qualitative analysis	104
8.4	Statistical Testing	105
8.5	Answers to Research Questions	105
9	Conclusion and Future Work.....	108
9.1	Conclusion.....	108
9.2	Future Work	111
	References.....	112
	Sample Summaries.....	122

List of Tables

Table 2.1: Strengths and Weaknesses of Term-Based Methods.....	13
Table 2.2: Strengths and Weaknesses of Clustering-Based Methods.....	19
Table 2.3: Strengths and Weaknesses of LSA-Based Methods.....	24
Table 2.4: Strengths and Weaknesses of Graph-Based Methods	31
Table 2.5: Strengths and Weaknesses of Miscellaneous Methods	34
Table 4.1: Characteristics DUC 2004 and Recent News Articles at a glance	71
Table 4.2: Ablation study representing the effects of different pre-processing phases on Grapharizer.	73
Table 4.3: Comparison of Grapharizer With State-of-The-Art Graph-Based Methods	75
Table 5.1: ROUGE Scores of RST Summarizer.....	82
Table 6.1: Comparison of Grapharizer with State-of-The-Art Systems	87
Table 7.1: Comparison of Grapharizer with Machine Learning Methods.....	94
Table 8.1: Rouge scores of the SOTA systems, devised systems, and ML algorithms.....	101

List of Figures

Figure 1.1: Taxonomy of automatic text summarization.....	3
Figure 1.2: Thesis organization.....	7
Figure 2.1: Clustering-based multi-document summarization.....	15
Figure 2.2: Latent Semantic Analysis (LSA) based multi-document summarization.	20
Figure 2.3: RST-Based multi-document summarization	26
Figure 2.4: Graph-based multi-document summarization	30
Figure 2.5: Challenges of EMDS.....	37
Figure 3.1: Example of Multi-Word Expression mapping [46].....	43
Figure 3.2: Module 1- Graph based Extractive MDS: pre-processing	45
Figure 3.3: Module 1- Graph based Extractive MDS: The processing.....	45
Figure 3.4: Module 2- RST based Extractive MDS.....	47
Figure 3.5: Module 1- Graph based Extractive MDS: Machine Learning	48
Figure 3.6: Research Strategy of Proposed Framework	50
Figure 4.1: The Overview of the Grapharizer.....	58
Figure 4.2: The Research Design of Grapharizer	63
Figure 4.3: Working of the Cross() for synonym mapping	65
Figure 4.4: Word graph construction from the given sentences.	67
Figure 4.5: The core model of Grapharizer	69
Figure 4.6: Grapharizer vs SOTA techniques for ROUGE 1	76
Figure 4.7: Grapharizer vs SOTA techniques for ROUGE 2	76
Figure 4.8: Grapharizer vs SOTA techniques for ROUGE L	77
Figure 4.9: Grapharizer vs SOTA techniques for ROUGE W.....	77
Figure 4.10: Grapharizer vs SOTA techniques for ROUGE SU	78
Figure 5.1: User Evaluation of RSTSummarizer	83
Figure 6.1: Rouge 1 comparison of Grapharizer with SOTA methods	88
Figure 6.2: Rouge 2 comparison of Grapharizer with SOTA methods	88
Figure 6.3: Rouge L comparison of Grapharizer with SOTA methods	89
Figure 6.4: Rouge W comparison of Grapharizer with SOTA methods.....	89
Figure 6.5: Rouge SU comparison of Grapharizer with SOTA methods	90
Figure 7.1: Rouge 1 Comparison of Grapharizer with ML variants.....	96
Figure 7.2: Rouge 2 Comparison of Grapharizer with ML variants.....	96
Figure 7.3: Rouge L Comparison of Grapharizer with ML variants	97

Figure 7.4: Rouge W Comparison of Grapharizer with ML variants	98
Figure 7.5: Rouge SU Comparison of Grapharizer with ML variants.....	98
Figure 7.6: Rouge SU4 Comparison of Grapharizer with ML variants.....	99

List of Abbreviations

AI Artificial Intelligence
ANN Artificial Neural Network
ATS Automatic Text Summarization
BLEU BiLingual Evaluation Understudy
BOW Bag of Words
ChatGPT Generative Pre-trained Transformer
DUC Document Understanding Conference
EDU Elementary Discourse Unit
EMDS Extractive Multi-Document Summarization
HDP Hierarchical Dirichlet Process
LDA Latent Dirichlet Analysis
LSA Latent Semantic Analysis
MDS Multi-Document Summarization
ML Machine Learning
MLR Multivariate Linear Regression
mTurk Mechanical Turk
NLP Natural Language Processing
NYT New York Times
ROUGE Recall-Oriented Understudy for Gisting Evaluation
RST Rhetorical Structure Theory
SLR Systematic Literature Review
SOTA State of the art
SVD Singular Value Decomposition

SVM Support Vector Machine

TF*IDF term frequency * inverse document frequency

TAC Text Analysis Conference

YAGO Yet another good ontology

Chapter 1

Introduction

1 Introduction

Since the emergence of computers, the reliance of individuals and companies over computers is increasing with tremendous pace. With the invention of the Internet, the reliance became more evident. The amount of data and information stored on disks started increasing. Today, the information retrieval from such huge amount of data is the major task. This problem is also named as information overload [1] – [4]. Information must be presented in a concise style in order to be accessed quickly. The Automatic Text Summarization (ATS) system is one approach to resolving this problem. ATS has its roots deep in the history of Natural Language Processing (NLP) for more than fifty years [5], [6]. Text summarizing is the process of extracting information from sources such that important details are not lost, and the redundant material is avoided [1], [2]. The Internet allows access to a massive volume of documents on a wide range of topics, with a high level of redundancy. The rapid expansion of the Internet has resulted in an explosion of electronic documents, making it difficult for readers to extract information from a large number of relevant and comparable materials. It is difficult for user to extract required information from a document or collection of documents. The user might not be able to read all the documents carefully. The information extracted can be false or incomplete that might cause trouble in future and the persons might have to start the work again from scratch.

Text Summarization saves the time and effort of user instead of reading the whole document user can simply get the information from the generated summary. Text resources are becoming increasingly abundant in the era of big data. It provides a summary that allows us to quickly comprehend the entire text while using the fewest words possible. The goal of text summarization is to create a condensed version of the original texts by shrinking the size of the papers while keeping the essential qualities.

Text summarization can be divided into two broad categories [7]:

1.1 Single Document Summarization

Single document summarization is the process of extracting most important information from a document in a concise format to ease the readability [7], [8]. It condenses only one document into a summary. The user can gain the required information in limited time. A document may contain repeated information, summarization process will also remove this issue.

1.2 Multi Document Summarization

Most of the time, same events can be covered from multiple sources, so we are presented with number of documents to gain an insight into it [5]. The information is to be extracted from many documents. In such case, the gist of the content is to be gained from multiple documents. The process of summary generation in such a case is known as multi document summarization (MDS) [7], [8]. MDS condenses a document collection into a single concise summary. It is a computer-assisted process for extracting information from many documents on the same subject. Information reports that are both succinct and thorough are created using MDS. It is used to extract the most important information from a collection of documents to produce a compact summary. It improves information services by generating brief and useful reports from many documents. In MDS, the goal is to provide a compression of the content of the entire input set given a set of documents as input.

Due to the increasing demand of text summarization, it has been even used to facilitate people with different medical issues. One such example is used by Barbu et al. [9] for the people with autism spectrum disorder (ASD). Similarly, the researchers from other languages also benefited from the techniques used for multi document summarization by using it for summarization in their own language, for example, Oufaida et al. [10] used Minimum Redundancy and Maximum Coverage algorithm (mRMC) for Arabic text.

Summarizing multiple document has proven to be far more difficult than summarizing a single document. This challenge results from the unavoidable thematic diversity found in a big collection of documents. The content in a group of documents can be succinctly described using a multi-document summary, which also makes it easier for people to comprehend the group of documents [11].

Summary can be Abstractive or Extractive based on the method of summarization. An extract is a summary consisting of sections of text taken verbatim from the source. Whereas an abstract is generally characterized as a summary built up of ideas obtained from the source, which are then reinterpreted and presented, in a different form [12].

Therefore, we can divide the summarization into two types:

i) Abstractive Summarization

It involves the summarization method in which the summary is generated by including the gist of the text in different words from the input file(s). This method involves deep techniques of

NLP [4], [13]. The language of the content is manipulated before presented in summary in this technique.

ii) Extractive Summarization

This approach ranks and prioritizes the most important passages in the document(s), and the most important sentences are combined to form the summary [14].

The desire to gain a thorough understanding of a subject without dedicating long period of time to it is the most recent trend. The requirement of meeting this difficulty prompted thorough research into extractive text summarization techniques. This method is not only faster than abstractive approaches, but it also guarantees a higher level of accuracy due to direct retrieval of texts. Because of the summarization technique, the reader does not have to be concerned about an incorrect interpretation of the text. Readers can also get to the core of the topic by reading specific terms in their raw form.

Additionally, the extractive summarization techniques can be divided into generic or query-based (given a query or not) and supervised or unsupervised (with/without a training set) approaches. Moreover, a generic summarizing relies on creating a summary of entire documents' important points, but query-based summarization creates a summary according to the topic of the user's question [7], [8], [13], [15] – [20]. Figure 1.1 represents the overall taxonomy of ATS.

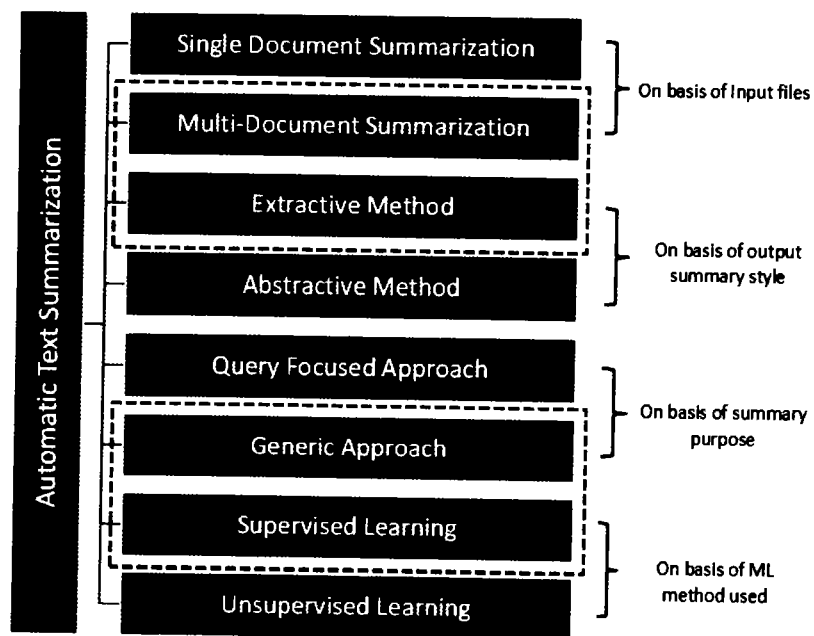


Figure 1.1: Taxonomy of automatic text summarization

1.3 Research Motivation

Methodology wise, extractive summarization is divided into Cluster based techniques and Graph based techniques [6]. The cluster-based method which was pioneered by Radev et al. [21]. It uses cluster centroids with top ranking tf-idf, or term frequency-inverse document frequency, for clusters' representation. The summary is then created by choosing the sentences from each cluster that most closely resemble these centroids. The k-means algorithm, which is based on partitional clustering, is a popular clustering method [21], [22]. In its attempt to express diversity and minimize redundancy throughout multiple documents, cluster-based approaches have been successful. You may see some of the publications that use the advantages of the clustering approach to build summaries in [23–26]. The graph-based approaches build a graph-based model, and then select sentences by means of voting from their neighbors using ideas like the well-known PageRank algorithm [27–29]. The links between sentences provide support for the fundamental theory of the graph-based technique. These connections are based on how closely the sentences resemble one another. The most commonly used similarity metric, like in the majority of graph-based approach literature, is the cosine similarity measure [30]. For the purpose of choosing a summary sentence, sentences with high similarity weights (in comparison to other sentences in the documents) will be ranked at top. Google's PageRank [29] is a well-known graph-based ranking algorithm that has historically been applied to social networks and Web-link analysis. Since it could find prestigious sentences throughout the documents, the graph-based method gained popularity for tasks involving the summarizing of multiple documents [31–34]. But if we examine the fundamental idea behind the graph-based method, the 'relation' between phrases is decided solely on its measured similarity value and not on its type of relationship [6].

Numerous approaches based on latent semantic analysis [35] and non-negative matrix factorization [36, 37] have been presented in order to take into account the latent semantics of document content. Additionally, certain ontology-based methods [38, [39]] have been applied to create summaries utilizing lexical semantics [15]. Ensemble based technique was also tested [40] for MDS and compared with four state-of-the-art techniques. While Rhetoric based summarization was also considered for the same purpose [41].

The related work is discussed categorically in detail in chapter 2.

The motivation behind this thesis was to devise a framework for EMDS (extractive Multi-Document Summarization) systems in which the effect of machine learning over the process

of MDS were examined. The intentions were to test the proposed techniques of graph-based system and RST based system with various popular systems of extractive MDS namely ontology-based system, term-based system, and pattern-based system. This framework is beneficial for evaluating the effects of machine learning over the extractive MDS as well as comparing the state-of-the-art systems with each other to conclude the best solution for extractive MDS.

The process of summarization, whether it's a single document or multiple documents, leads to compromised grammaticality. This issue worsens when it comes to extractive summarization, where chunks of sentences are selected from the given document(s) to create a summary. As a result, the summaries often lack information, diversity, and contain redundant content. Hence, there is a requirement for an extractive Multi-Document Summarization (MDS) solution that can generate grammatically correct summaries that are informative, diverse, and non-redundant. Some open issues in the field of Extractive MDS are stated in chapter 2.

In order to create and implement an extractive MDS framework that tackles issues such as lack of information, diversity, redundancy, and grammaticality, we must utilize various solution strategies. These strategies include topic modeling, synonym mapping, pronoun replacement, and handling multi-word expressions. To achieve this, we will apply these solution strategies to both graph-based and RST-based extractive MDS techniques. Subsequently, we can compare our results with the current state-of-the-art approaches in extractive MDS, which encompass term-based, ontology-based, and close pattern-based techniques. It is worthwhile to incorporate the benefits of machine learning in order to generate a desired summary.

1.4 Thesis Contribution

The contributions of this thesis are as follows:

- A comprehensive review of existing literature was undertaken to examine the present body of evidence regarding EMDS.
- A system for extractive multi-document summarization was designed and implemented based on graph theory principles.
- A learning based system for extractive multi-document summarization, utilizing graph theory, was developed and implemented.
- A system for extractive multi-document summarization, based on rhetoric structure theory, was devised and implemented.
- Implemented the state-of-the-art baseline systems for the purpose comparison

- implemented term-based EMDS system.
- implemented YAGO ontology-based EMDS system.
- implemented Closed Pattern based EMDS system.
- A comparative evaluation was conducted between extractive multi-document summarization systems employing graph-based and rhetoric structure theory-based approaches, and state-of-the-art baseline systems such as term-based, ontology-based, and closed pattern-based systems. The evaluation utilized ROUGE evaluation metrics as well as other quality parameters including informativity, coverage, grammaticality, and redundancy mitigation. Multiple datasets were used for the evaluation process.

1.5 Thesis organization

This thesis is organized in nine chapters, as shown in figure 1.2, as per the following arrangement:

Chapter 1 provides introduction to the thesis. Chapter 2 mentions the literature that has been surveyed and that indicates to the formulation of the research problems. Chapter 3 presents the detailed research methodology to mention the settings in which the research studies were conducted. Chapter 4 represents the first module of the solution which involves graph-based technique for summary generation, namely Grapharizer. Chapter 5 presents the second component of the framework that finds out the RST based extractive multi-document summarizer, named RSTSummarizer. Chapter 6 presents the implementation details of SOTA baseline techniques from different domains of extractive MDS. It is worth mentioning that the SOTA techniques were implemented in the same settings in order to see the effects for better comparisons and evaluation of the methods we devised. Chapter 7 provides the effects of machine learning techniques over the summarization process. Chapter 8 compares and discusses the results in order to evaluate the performance of extractive MDS over different techniques, on different datasets, using quantitative and qualitative evaluation metrics. Chapter 9 finally concludes the research and gives future directions.

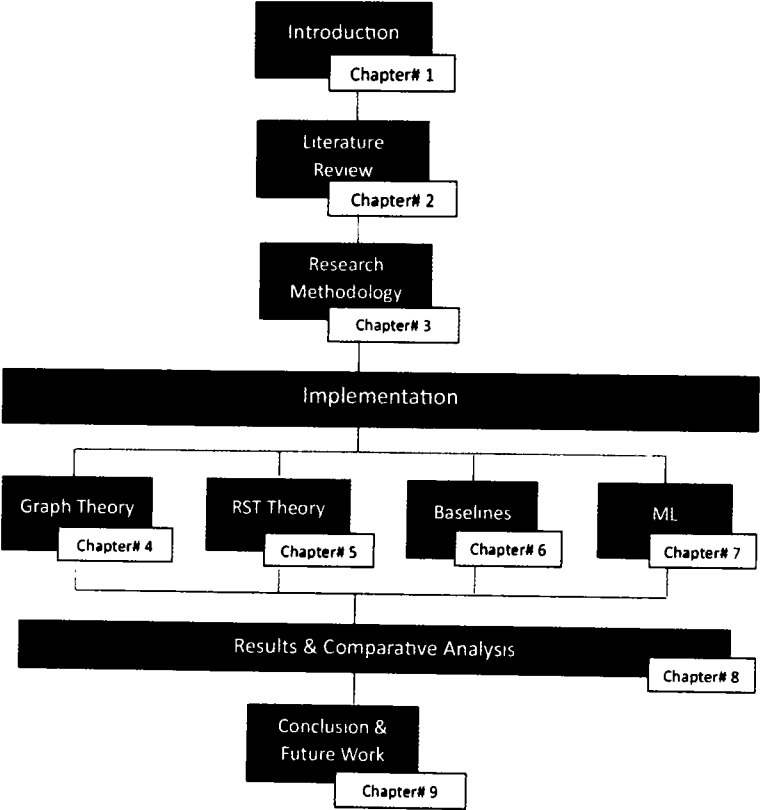


Figure 1.2: Thesis organization

Chapter 2

Related Work

2 Related Work

Increasing reliance and storing of data and information is generally associated with information overload [1]–[4]. Information must be presented in a more compact style in order to be accessed with the least amount of time. One answer to this problem is ATS, which has a long history in text summarization going back more than 50 years [5], [6]. Text summarizing is the process of extracting information from a text so that the important information is not lost and the redundant parts of the original format are not included in the final summary [1], [2].

As text summarization eliminates redundant data from digital documents, it has been used to facilitate computer use by people with different medical disabilities. For example, text summarization was used for people with ASD by Barbu et al. [9]. Similarly, researchers from languages other than English have also benefited from the techniques used for multi-document summarization by using it in their respective languages. One such example is Oufaida et al. [10], who used Minimum Redundancy and Maximum Coverage algorithm (mRMC) for Arabic text.

Single-document summarizing and multi-document summarization are the two main categories into which text summarization can be classified [7]. Single-document summarization is the process of extracting the most significant information from a document in a concise format for ease of readability [7], [8]. In situations where the information is dispersed among numerous sources and documents, multi-document summary is used. For instance, the same information could be covered by several sources, making it possible to often access a variety of documents to get insight into the same event [5]. In this way, a multi-document summary represents the data in a collection of documents and aids users in understanding the main points of those documents [7], [8]. A multi-document summary helps people comprehend the group of documents by representing the information found there. [11].

Even when the accessible single document is very vast in size, the work of multi-document summarizing is far more difficult than single-document summary. This difficulty is attributed to the inevitable diversity of themes within a large set of documents.

A summary can be Abstractive or Extractive, depending on the method of summarization. Generally, an abstractive summary consists of concepts and ideas abstracted from the source document(s) and then represented in preferably different words. This approach necessitates a

profound comprehension of the content's meaning. Semantic representation and natural language generation [4] are two broad categories of NLP [13] that deal with abstractive summarization. These use a variety of methodologies, including methods based on information items, multimodal semantic models, and semantic graphs [50].

Units of text that have been taken verbatim from the source document(s) are referred to as extractive summaries [12]. With this approach, the summary is created by ranking and combining the key phrases from the documents in question [14]. The several types of extractive summarization techniques include query-based, generic, supervised, and unsupervised approaches. Generic summarizing relies on creating a summary of the documents' main points, whereas query-based summarization creates a summary focused on the topic of the user's question [7], [13], [15]–[20].

In order to gain a broader picture of research in this field, we have performed a systematic survey of the literature on extractive techniques of MDS. The survey may serve as a starting point for uninitiated researchers to identify gaps in current research.

2.1 Literature Review

Many multi-document summarization systems are available in the literature. Methodology-wise, extractive summarization is divided into Cluster-based techniques and Graph-based techniques [6]. The cluster-based method was first presented by Radev et al. [21] basic idea of which was to group similar sentences from the document(s) into clusters, and then choose the most salient sentences from each cluster to compile a summary of the document(s) [21], [22]. Radev et al. used tf-idf based features in k-means clustering algorithm to group similar and salient sentences together. Tf-idf scores the importance of words (or “terms”) in a document based on how frequently they appear in multiple documents. If a term is frequent in all the documents, tf-idf ranks it low, assuming that it is not a salient term of a specific document; instead, it is a common term. Tf-idf helps to filter out closed-class words that are used frequently in a language but are not representative of the meaning of the document. The summary produced by cluster-based approaches incorporates a variety of information from texts while at the same time minimizing data redundancy. In [23]–[26], well-known cluster-based summarizing methods are presented. The graph-based techniques [27], [28] utilizes the concept of the well-known PageRank algorithm [29], which was traditionally employed in social networks and web-link analysis. They build the sentence graph, and then their neighbors vote to select a sentence for the next vertex. The fundamental graph-based theory is maintained

by the links between sentences existing based on some similarity values calculated by some techniques (like cosine similarity measure [30] between sentences. The calculation of sentence similarity takes into account other sentences in the documents. The most effective summary sentences are those with high similarity values. In multi-document summarization, the graph-based technique is utilized to find important sentences across multiple documents [31]–[34]. When using the graph-based technique, sentences are connected according to a similarity value rather than a relationship type [6].

While taking the latent semantics of the contents of the documents in a view, several methods are devised based on latent semantic analysis [35] and non-negative matrix factorization [36], [37]. Similarly, keeping in view the lexical semantics [15], ontology-based approaches [38], [39] have been used to produce summaries. Ensemble-based technique was also tested [40] for multi-document summarization, while Rhetoric based summarization has also been considered for the same purpose [41], [44].

Based on the literature studied, there are several widely used extractive summarization methods. Some of the categories are stated as follows:

2.1.1 Ontology-Based Methods

Ontologies are formalized depiction of the most unusual concepts related to a specified knowledge domain and different corresponding relationships. They are used in numerous research fields, including user-generated content analysis, e-learning framework development, video analysis, and image analysis. Recently, the use of ontologies is increased by the research community [16], [38], [39] due to its promising results in various fields, specifically in document summarization. It helps identify important sentences from the documents to generate a summary by incorporating ontological knowledge. Ontologies are used to show the document set's critical concepts and their correlation with the user query by avoiding ambiguities.

An ontology-based approach was proposed by Baralis et al. [38], called YAGO summarizer, which used Wikipedia for mapping of words to non-ambiguous ontological concepts called entities. YAGO summarizer selects sentences from a document as per previously assigned entities.

This technique's achievement is the use of ontology of a domain, which consequently eliminates the problems of synonymy and polysemy in MDS. The limitation of ontology-based

approaches is that the ontologies are domain-specific. Similarly, much of the efforts are needed to develop an ontology of some domain [15].

2.1.2 Term-Based Methods

The term-based methods commonly implement the bag-of-words (BOW) model to calculate the weight of a term using the tf-isf weighting model and some variants of this scheme.

Oliveira et al. [4] presented a analysis of 18 shallow sentence salience-scoring methods side by side to determine each sentence's importance in extractive single and multi-document summarization. Numerous experiments were performed to evaluate the performance of these sentence-scoring methods separately and utilizing various combination techniques over the news domain datasets of CNN Corpus and DUC 2001-2004. The sentence scoring techniques used various combinations of features like word frequency, word co-occurrence, upper case, TextRank, tf-isf, sentence resemblance to the title, position of the sentence, length of the sentence, centrality of the sentence, proper noun, open relations, numerical data, noun and verbal phrases, named entities, lexical similarity, cue-phrases, aggregate similarity, and bushy path. These scoring techniques were used as input features for different machine-learning algorithms provided by Weka toolkit, like AdaBoostM1, J48, K-nearest Neighbours referred as IBK, Multilayer Perceptron, Multinomial Logistic Regression (Logistic), Naive Bayes, Random Forest, Random Tree, Radial Basis Function Network (RBFNetwork), and Support Vector Machines using Sequential Minimal Optimization (SMO). The state-of-the-art techniques for SDS selected were Autosummarizer, Classifier4J, and HP-UFPE Functional summarization, along with the best performing participants of DUC 2001, 2012, while for MDS, the state-of-the-art systems were ICSISUMM, Greedy-KL, LLRSum, ProbSum, Sume, as well as the best performing participants from DUC 2001-2004 competition. It was observed that in combination with state-of-the-art, these techniques produce better results, but the standalone performance of these techniques is a bit compromised.

In order to solve the optimization problem, another method known as Maximum Coverage and Less Redundancy (MCLR) [7] models MDS as a quadratic boolean programming problem. This method maps the objective function using a weighted mixture of the content coverage and redundancy objectives [20].

A bottom-up approach for arranging the sentences was presented by Bollegala et al. [42]. They developed criteria based on chronology, topic-closeness, precedence, and succession to determine the association between two sentences and determine their order [15].

The sequence in which the information appears in the created summary is quite important. [43] iterates this need by first extracting the most important sentences from the given documents. Five criteria—chronology, probabilistic, topic-closeness, precedence, and succession—are used to establish this extraction. The meaningful extracted sentences are then arranged to add to the beauty of the summary. This ordering is done by using human-annotated summaries in the system. Once the system learns the best combinations, the model is tested on the automatically generated summaries. The proposed sentence ordering procedure operates on pair-wise comparisons of sentences to determine the overall ordering. This is done using a greedy search algorithm that avoids the combinatorial time complexity, which is typically associated with total ordering tasks. This helps in quick sentence-ordering in more extended summaries; therefore, this approach is feasible for real-world text summarization systems.

Nasir et al. [57] used a measure of semantic relatedness, named Omiotis, to construct a flattening matrix and a kernel for semantically adjusting the BOW illustration. Omiotis is made from the thesaurus and WordNet (word dictionary), which handles the problem of synonymy and polysemy. Omiotis works on sense-related measure SR. It uses the BOW approach by embedding Omiotis into a semantic kernel. The recommended measure includes the tf-idf for producing a semantic kernel by combining the semantic and statistical information related to the text. It handles the word synonymy and polysemy problems. The Latent Semantic Analysis, discussed in detail in sub-section 2, helps handle the problem of synonym, but polysemy is yet to be resolved.

Document summaries have also been created using Bayesian topic modeling [61]. The document's sentences have numerous embedded topics that are not focused in the majority of the summarizing strategies. More importantly, it emphasizes the hidden embedded topics present in sentences to generate an appropriate and precise summary. On basis of this method, it can be concluded that topic modeling helps in understanding the context by selecting the appropriate sentence, which would help generate an effective summary by makes use of both text document and the word sentence relationship.

The textual entailment relations and sentence compression by the Knapsack problem was used by Naserasadi et al. [74]. It is used to address the extractive MDS problem. It first ranks the sentences by tf-idf method and then calculates the entailment scores of the selected sentences. The sentence's final score is calculated, and then the sentences are compressed through a greedy dynamic programming approach for the Knapsack problem. This technique gives 2%

improvement in the query-based approach of summarization, while for the generic summary, 5% improvement is recorded. The knapsack problem is one of the optimization problems. Here, sentences are considered problem items, and their values are calculated by “production” of entailment score and tf-idf value. ROGUE 1, ROGUE 2, ROGUE SU4 are used for evaluation, while the datasets selected were DUC 2007 for query based and MultiLingPilot 2011 for generic summarization.

The comprehensive comparison of all the term-based methods is presented in Table 2.1.

Table 2.1: Strengths and Weaknesses of Term-Based Methods

Sr.#	Research Study	Working	Results & Evaluation
1	Naserasadi, A , Khosravi, H., & Sadeghi, F (2019)	Sentences are ranked, and then entailment scores are calculated and then finally compressed using 0-1 Knapsack problem	Gives 2% improvement in the query-based approach, while 5% improvement is recorded for the generic summary Efforts are required to decrease the complexity of the algorithm
2	Oliveira et al , 2016	Eighteen shallow sentence scoring techniques are compared on different methods in the news domain. It is applied on SDS as well as MDS using an extractive method of summarization	Individual results of these sentence scoring techniques are not promising. When combined with state-of-the-art methods, these techniques show comparable results
3	Alguliev, Aliguliyev, & Hajirahimova, 2012	MCLR technique is presented. Sentences are scored as per features, and then prominent sentences are compared with each other. Unique ones are included in the summary For optimization, a modified Differential Evolution algorithm is used	Individual results of these sentence scoring techniques are not promising. When combined with state-of-the-art methods, these techniques show comparable results.
4	Bollegala et al , 2012	Probabilistic criterion is added to the work presented in Bollegala et al., 2010	It was only tested on the Japanese News dataset. Furthermore, testing is required on benchmark datasets
5	Nasir et al., 2011	Omiotis measure of sense relatedness is used alongside the BOW approach to handle synonymy, polysemy, and word semantic relatedness problems	In pre-processing of the data, stemming was missed Due to grammatical considerations, stemming would have eliminated the same root terms used in different ways.
6	Bollegala et al , 2010	A sentence association and ordering technique is presented based on the criteria of chronology, topic closeness, precedence, and succession	The algorithm is tested on a dataset of Japanese newspapers However, it needs to be tested on standard benchmark datasets
7	D. Wang et al , 2009	It discusses how to interpret phrase context using the Bayesian method of topic modeling. For sentence ranking, tf-idf is employed	LDA uses exceptional maximization that increases the complexity and slows down optimization Pre-processing of the data using deep natural language analysis is missing

The further categories of the term-based method are as follows:

2.1.2.1 Clustering-Based Methods

Based on a set of features, clustering-based methods compute the similarity between sentences, also known as the salience of sentences, to rank them. MEAD [21] is an example of a

clustering-based method that is used for sentence extraction. This task is done with three parameters, namely, the value of centroid (sentences' average cosine similarity to the remaining sentences in the documents), positional value (documents contain N sentences, leading sentences is given 1 as a score and for each sentence the score decreases with the ratio of $1/N$), and finally the first-sentence overlap (the cosine similarity of a sentence with the first sentence in the same document). The three parameters are linearly combined and assigned equal weights. Figure 2.1 describes the clustering-based methods in detail.

Density-Peak Clustering Sentence proposed by Zhang et al. [53], calculates the sentence representativeness score and diversity score. It first calculates the sentence similarity matrix by dividing documents into sentences and then removing the stop words. After that, the sentences are represented as a bag of words, and a cosine comparison is calculated. The Boolean system is used to assign weights to the sentences, and the representativeness score is calculated. Representativeness score describes the sentence that is important in the document. After that diversity of the sentences is calculated. Diversity score condenses the redundancy, which was the task of the post-processing unit. It is calculated by computing the minimum distance between some sentence i and the other sentence having the highest diversity. Length score helps to make the sentence length shorter. Real length is the number of words in a sentence, whereas effective length refers to the number of unique nonstop words in a sentence, i.e., the sum of unique words. The squatter sentences with better representativeness are extra ideal over those with long length. Experiments were done on dataset of DUC 2004. It is therefore confirmed that the density peaks gathering method can effectively handle multi-document summarization. However, this work is at an initial stage and is open for further research inputs.

Wang et al. [63] presented Density-Peak based clustering technique for generic extractive multi-document summarization. The benefit of Density-Peak's technique is that it does not demand to set the number of desired clusters in advance and is handled at run time. In clusters, sentences are ranked using Integrated Score Framework, and salient sentences are selected to be part of the summary using dynamic programming. This technique performed well in the ROUGE SU4 matrix for summary evaluation, while in ROUGE 1 and ROUGE 2, its performance was not better than other techniques. Similarly, this technique did not handle the problems of synonymy and multi-vocal words in this work.

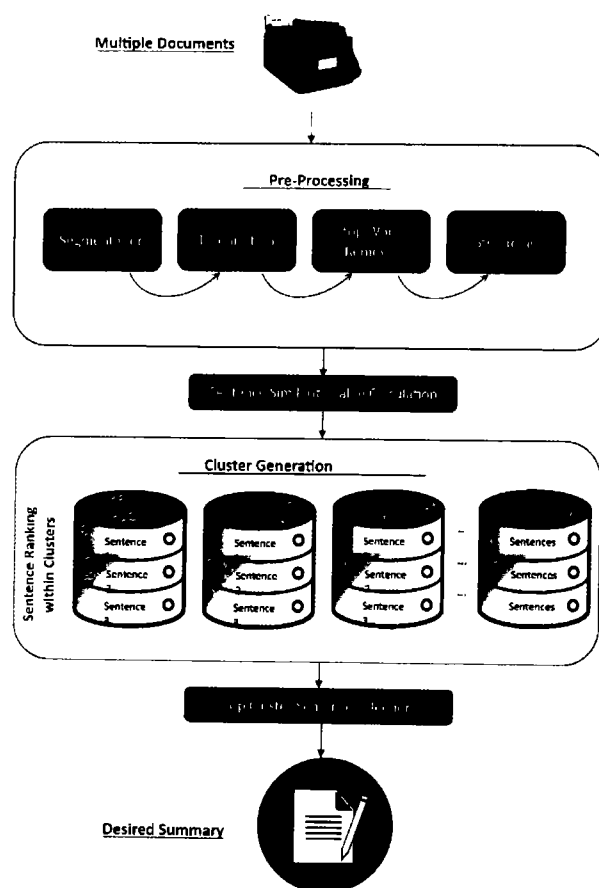


Figure 2.1: Clustering-based multi-document summarization

Nagwani et al. [55] worked on Big Data Analysis and presented the summarization of large data available in it. This is accomplished using topic modelling and semantic similarity clustering. This work is done in four stages. First, text clustering is used on the documents to create clusters via K-means so that similar documents contribute to the summarization task. In the second phase, Latent Dirichlet Allocation (LDA) creates the topics from given sentences. In the third stage, frequent word generation is done by sending the topic words (terms) produced from the LDA to the summarizer, then mixed up and transmitted to the mappers. Topic-terms frequency is computed, and frequently occurring terms are produced. After that, semantically similar terms for the frequent terms are produced with the help of WordNet. In the last stage for each document, sentence filtering is performed based on semantically identical words and frequent words. Sentences are picked from every document for frequently occurring words and their semantically similar words to constitute the summary. Duplicate sentences are removed, and a summary is generated. The MapReduce implementation collects all values linked with the same key and combines them in the reducer. The result of the algorithm is obtained in the distributed file system, having a file per reducer. In the end, the sentences

containing the frequent terms are selected that will produce the summary of the given text. This is quite a detailed solution but a very costly one in the context of MDS. This work is done for big data analysis and inherits the drawbacks of the K-means algorithm, employing extensive external sources.

Christensen et al. [60] investigated hierarchical summarizing, in which the top-level sentences offer a summary of the documents such that more information can be retrieved by guiding them into sentences. This separates parent-child consistency and provides pertinent information based on user attention so that a user with a certain interest can delve further into the content of interest. This is how, the root sentence gives a general overview of the summary. By selecting an additional sentence of the summary, it gives more detail about the occasion. If the third sentence is selected, it will further provide information to go into depth and gain more details. In this fashion, each non-leaf node offers additional information about the leaf nodes, i.e., a child contributes additional information about the parent. A method called SUMMA summarizer, used by Christensen et al. [60], was used to summarize sentences. SUMMA employs articles before grouping the sentences into a cluster with a time-related objective function, which improves the information's salience and coherence. In hierarchical summarization, input is a set of related documents. There is a budget for each summary. The output is hierarchical summary and set of summaries. Child summary gives more details to the information, i.e., events or any other background. Each summary should have coherence which comprises of parent-child consistency and intra-cluster coherence. Initially, the quantity of information is less, and the user directs it as a topic of concern. Process of the summary generation is shortened into two parts. The first one is to create clusters and the second is summarization inside the clusters. Hierarchical clustering results in the clustering based on chronology. Then summarization of the gathered documents cluster is performed chronologically. Clustering algorithm is used recursively to choose the number of clusters which are time stamped prior to the gathering. Sentences are then parsed with Stanford parser. Documents are drawn to the topic by a sentence to topic value called salience. It adds the saliencies of individual sentences. Training of dataset was done with linear regression classifier, which is also used for identification of redundant sentences. The features include shared noun counts, sentence length, tf-idf, cosine similarity, and timestamp difference. In this regard, two types of coherence are required here, one is the parent-child coherence, and the other is coherence within each cluster. Therefore, an approximate discourse graph (ADG) is used for calculating coherence. In parent-child coherence, the user will move from the parent

sentence to the child sentence, so there must be a proper link among parent-child sentences, and the sum of positive weight from parent sentence to a child will be displayed in ADG. In intra-cluster coherence, the summary is deemed acceptable if it has positive evidence in ADG. For calculating the quality of summary produced, a function is used that combines consistency, salience, and redundancy. Therefore, the number of sentences in summary must match the non-leaf cluster. The issue is that it deals with redundancy and budget as hard constraints while considering coherence and salience as soft constraints. It is based on timestamps and is location-focused.

Clusters with random shapes can simply be noticed employing local density methods. It adapts the K-medoids technique. In an algorithm by Rodriguez et al. [62], cluster centers are enclosed by low local compactness neighbors. They have comparatively large space from any points with a higher local density. For each specific point, two modules are calculated: local density, and the other is the points with higher density. In local density, those points that are not close to D_c are cut down. D_c is the value that shows that point that is not closer to the distance between the data points D_{ij} will be removed.

The other parameter is computed by discovering the least distance of point i from all the other points with higher density. If this parameter has a large irregular value, then it is measured as the cluster centre. The algorithm has no noise cut-off. First, the border region of the cluster is defined. These will be the points that are assigned to the cluster. These points have a distance D_c from points that belong to other clusters. For each cluster, the point with the highest density is selected from the border region. The points above this value are considered part of the cluster core, and the other points are considered noise. This algorithm gets the position and shape of the clusters, which have even different densities. From a large number of points, reduced samples are gained, and cluster assignment is performed in it. The wrong classified points' fraction remains below 1 percent, even for small samples containing 1000 points. In some cases, the datasets with a small number of points might be affected by significant statistical errors.

Argumentative Zoning was also used for extractive summarization in the scientific domain [66]. A trained classifier is used along with a feature-based clustering technique. The classifier's job is to create a preliminary candidate set of sentences to be included in the summary. The sentence cluster is used for identifying groups of connected (similar) sentences in that set created by the classifier. These groups are then used to generate the final summary.

Clustering improves the quality of summary by removing redundancy from the candidate set. Sentences from training articles are pre-processed, labelled, stop words are removed from sentences, and lemmatization is done. After that, sentences are represented as a feature vector for the training of the classifier. The compression ratio and the number of clusters are threshold values set by the user. After classification, cluster generation is used for summary generation. The classification uses set A to be a set of sentences in the abstract of papers and set M to be a set of sentences in the paper's main body. Using sentences in sets A and M, the classifier is trained to generate sentences in set C, which is a set of sentences in summary. The sentences in set A are positively labeled, while set M's sentences can be positive or negative. Here the non-traditional classifier-based method is used for training. Artificially generated data can be used to train the classifier. The features are verb features, tf-idf, citations and reference occurrence, argumentative zones, and locative features. It means that previous work is present at the start of the information and future work is present at the end. The summary is supposed to provide comprehensive information of related work of the topic and its methodology. After sentence classification, K-means clustering is used to remove redundancy and identify similar sentences, and the desired summary is generated using cluster centroid. Another sentence clustering method is to group the sentences having the same argumentative zones label for easy identification of clusters. As per user requirements, the system can produce full-document and customized-document summaries. Hence, the conclusion is that the argumentative zone helps in producing effective summaries of the scientific domain. The issue here is that positive and negative labelling of sentences is complex, and clustering and classification make it a little costly solution.

The method by Christensen et al. [60] yields the best Rouge-1 values among clustering-based algorithms, with a Recall of 0.67 on DUC 2004.

Table 2.2 summarizes the benefits and drawbacks of the clustering-based approach.

Table 2.2: Strengths and Weaknesses of Clustering-Based Methods

Sr.#	Research Study	Working	Results & Evaluation
1	B. Wang et al., 2017	It eliminates the need to tell in advance the number of desired clusters due to Density Peaks' use.	ROUGE 1 and SU4 give best results. However, it does not handle synonymy and polysemy problems which give lots of future directions for researchers.
2	Zhang et al., 2015	Representativeness, Diversity, and Length parameters are considered in the clustering-based method of MDS.	Performance at DUC 2004 is good. However, it does not handle synonymy and polysemy problems. Work is still in a preliminary stage, and refinements are in progress.
3	Nagwani, 2015	Summarization of extensive data available in BigData is performed using the MapReduce framework.	The algorithm designed in the study is evaluated on some legal documents. It would give us a better understanding of results if implemented on benchmark datasets. Similarly, the technique uses external resources extensively, which makes it an expensive MDS solution.
4	Christensen et al., 2014	Hierarchical summarization is presented where nodes provide additional information if we keep traversing until the leaf node is reached.	Redundancy and budget are treated as hard constraints and coherence and salience as soft constraints.
5	Rodriguez & Laio, 2014	Cluster borders are managed by calculating local density and high-density points.	In some cases, the datasets with a small number of points are affected by significant statistical errors.
6	Contractor et al., 2012	Sentences are labeled based on whether they appear in the abstract, main body, etc., of articles using Argumentative Zoning.	Clustering and classification are used together, which increases the cost. Positive and negative labeling of sentences is complex.

2.1.2.2 Latent Semantic Analysis (LSA) Methods

Gong and Liu [35] provided a method for ranking high-scoring sentences in the document collection using Latent Semantic Analysis (LSA) for the purpose of generating summaries. As shown in Figure 2, it creates a matrix of terms and sentences, where the columns show the weighted term-frequency vector of a sentence in the documents set. The latent semantic structure is then derived by using Singular Value Decomposition (SVD), which is a mathematical method to show the relationship among terms and sentences, on the input matrix. The document set is divided into various topics, and the sentences with the highest total weights across all the topics are chosen for the summary.

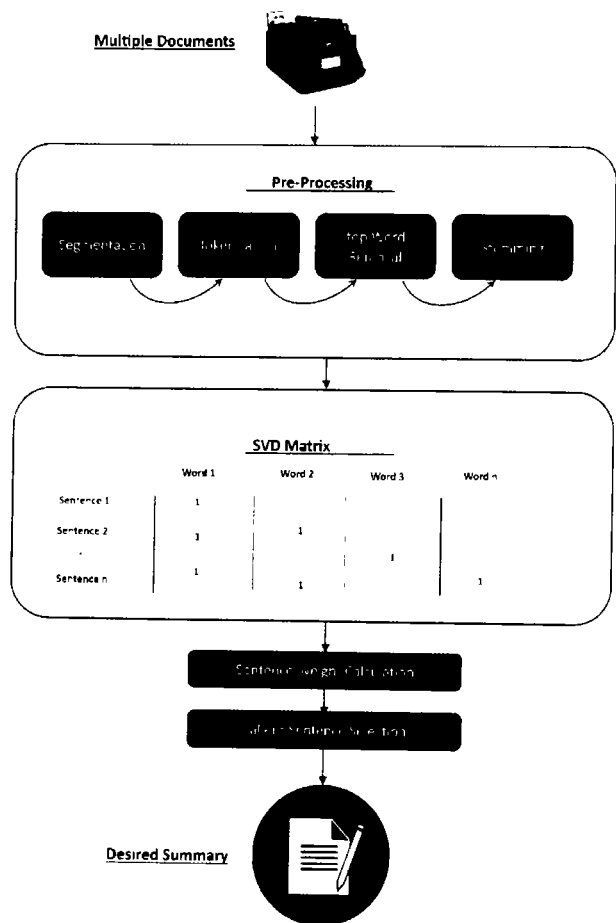


Figure 2.2: Latent Semantic Analysis (LSA) based multi-document summarization.

Ferreira et al. [45], who focused on sentence similarity and word order in their work, made another attempt to enhance sentence similarity algorithms. The authors claim that the following elements have not yet been taken into account by the scholarly community: The Meaning Issue: There are numerous ways to write statements with the same meaning yet written differently. Like the sentences “John is a handsome boy” and “John is a good-looking lad,” they have similar meanings if used in the same context. The Problem of Word Order: A text’s meaning can change depending on the order in which words appear, such as when phrases with the same mix of words are written in a different sequence “A killed B” and “B killed A” bring different implications. In their representation of sentences, Ferreira et al. [45] divided each phrase into three layers: (i) lexical, which performs lexical analysis, stemming, and stop-word removal; (ii) syntactic, which conducts syntactic analysis; and (iii) semantic, which primarily specifies the semantic role annotations. Additionally, a novel sentence similarity metric is presented in this study. The text semantics are obtained using semantic role annotation (SRA), which previously were obtained using WordNet. The three-layer sentence representation handles the problems of meaning and word order.

Marujo et al. employed the event-based approach [47]. Event data and word embeddings are used in MDS in this work. KPCentrality method that is already used in a SDS was extended for MDS. It was used in single layer as well as waterfall approaches. The single-layer approach generates a summary by adding the summaries of every input document at the end. On the other hand, the waterfall approach joins the summaries of every input document based on a timestamp of documents in a cascade style. Event information is used in the filtering stage and the improvement of sentence representation.

Maximum Marginal Relevance (MMR) [58] combines query-relevance and innovation criteria to remove redundancy. In the end, the dissimilarity is computed among the documents in the ranked list. MMR considers the relevant novelty, which can be calculated independently for the ranked documents. The text will only have high marginal significance if it is strongly related to the query and is least different from the earlier document. MMR helps to find out the relevant candidate documents quickly and to find the similarity between them. If the summary is to be found via relevant sentence extraction, it requires relevance and redundancy to be discovered out. In single document summarization, the documents are divided into sentences, cosine similarity is found out, and sentences are ranked for the summary. In MMR, the candidate selection score has two components; one is the relevance of the candidate with the user's query, and the other is a similarity of a selection of candidates with other candidates present in summary. These scores are computed in each iteration, and the algorithm stops after meeting specific criteria. MMR method works well for long documents as they have more repeated sentences. It is also suitable for the extraction of sentences about a similar topic in multi-documents. It helps in eliminating redundancy in query-relevant multi-document summarization. The problem with this algorithm is it does not help in reducing the global diversity, and it does not provide the facility scale to output with a larger size.

Lin et al. [59] characterize the interactive summarization technique by using the MMR algorithm, which helps the user select candidate sentences. This helps in generating highly interactive high-quality summaries than automatic summaries. Lin et al. [59] extended MMR [56] algorithm, which places users in a loop. The user is asked to select a sentence at each step that would be added to the summary. It gives the user a ranked list of sentences for selection. The evaluation vehicle for measuring the summarization algorithm's effectiveness is Complex, Interactive Question Answering (CiQA). CiQA consists of topics that have two parts, i.e., question template and narrative (description). Participants organize web-based QA systems with which NIST assessors interact. Each assessor interacts with the participant, after which

participants submit the final run. In experimentation, interactive MMR is executed after the initial run (standard run) is performed.

In interactive MMR, which is web-based, the user selects sentences at every step. For the final run, the output of the interactive run is combined with the output generated automatically. IDF is used to compute the relevance of each document in the experiment. Cosine similarity is used to eliminate redundancy. The interface consists of 3 components; question, current answer, and sentences ordered as scored by MMR. At each step, the user is asked to select the sentence which is then added into the current answer. F-measure is considered for evaluation measure. But the problem is it does not account for the sentences which have varying length. The weighted answer shows that how far relevant information is contained in the system response. Another downside of the solution is the human intervention which is necessary for the task but is hard and time-consuming.

Ozsoy et al. [64] tried to solve the shortcomings of previous approaches. The earlier methods first select the concept and then choose the sentences related to the concept, which is finally used in summary. Ozsoy et al. [64] used LSA-based methods on Turkish text and devised two techniques of sentence selection. The Cross method was used for sentence selection in the input matrix. This method's primary function was to determine that although sentences at the introduction and conclusion part tend to be more critical, there can still be some sentences selected that may cause noise in the matrices of LSA. Like previous approaches, the Vector Transpose VT matrix is used. The Cross method pre-processed this matrix before sentence selection. The average sentence score was calculated for each concept in VT matrix for every row. For cell values less than the average row score, they were set to zero, for these were sentences related to a topic somehow but not the core sentences. Then the length score is calculated for sentences. The sentences are selected based on higher values. To distinguish between the main topic and the subtopic, another method, named Topic method was proposed. It decided the main topic by creating a concept * concept matrix. This matrix added the cell values that were common among concepts. The strength value of concepts was calculated by considering each concept as a node and the similarity value of concepts * concept as edge score. Then values of concept in each row of this matrix are added to compute the concept's strength. Higher value concepts are considered as the main topics. Investigation on two data sets was performed, which was then related to human-generated abstract summaries. The issue here was the use of complex algorithms with SVD.

Data representation is complex in textual data, as it suffers different problems like uncertainty, imprecision, incompleteness, etc. This causes the problem of classifying the same sentences into different classes. Chatterjee and Yadav [67] used Fuzzy Rough Sets (FRS henceforth) based sentence similarity measures because FRS uses meanings of sentences. FRS is the combination of Fuzzy Sets and Rough Sets. Former deals with uncertainty through membership functions, while the latter with the help of lower and upper approximation of a set. Imprecision can be defined as something that is not precisely told. For example, consider the sentence “Ram is a man of medium height,” We have no idea about what a medium height stands for. On the other hand, uncertainty occurs due to polysemous words, anaphoretic pronouns, and structural ambiguity.

The lower and upper approximation is estimated as those that certainly belong to the concept make its lower approximation. In contrast, the elements that possibly can belong to the concept make an upper approximation. The technique for sentence similarity was tested on the SICK 2014 dataset, while for summarization DUC 2002 was used. Results reported on DUC 2002 were quite encouraging for ROGUE 1, ROGUE L, and ROGUE SU.

SDS is used for the extractive method [71]. The technique adopted is CNN, and it is tested on the datasets of CNN, Dailymail, and NYT. They compare the extracted sentences, keeping a particular focus on the grammar quality of the resultant sentences. Sentences are encoded using bidirectional Long Short-Term Memory (LSTM) and the Convolutional Neural Networks (CNN). Sentence representatives are identified and then aggregated with document representative that is encoded with bidirectional LSTM and CNN. Decoding is done with sequential LSTM. Greedy decoding is then applied at the testing phase for the nomination of the most likely sentence sequence. These selected sentences are then compressed by omitting some words or phrases to make them more concise. Compression rules and feed-forward networks facilitate the choice of deletion of words or phrases. The resultant summaries are evaluated at mTurk, Grammarly, and manual analysis. With the CNN dataset, the results were more promising as it contains compressed sentences already.

The methods based on LSA for MDS are presented in Table 2.3 for review.

Table 2.3: Strengths and Weaknesses of LSA-Based Methods

Sr.#	Research Study	Working	Results & Evaluation
1	J. Xu & Durrett, 2020	CNN and LSTM are used on extractive single-document summarization.	The resultant summaries are evaluated at mTurk, Grammarly, and manual analysis. With the CNN dataset, the results were more promising. It is recommended to apply compression over NYT and Dailymail datasets in order to get a better result there as well.
2	Chatterjee & Yadav, 2019	The Fuzzy Rough Set method is used to deal with sentence similarity and uncertainty issues within data.	Results reported on DUC 2002 were quite encouraging for ROGUE 1, ROGUE L, and ROGUE SU, while for the SICK 2014, the improvements can be made to get better results
3	Ferreira et al., 2016	Sentence similarity, word order, and meaning problems are handled in a 3-layered module of lexical, syntactic, and semantic layers.	It would be best to test on the standard dataset like DUC. The performance of semantic and syntactic measures showed promising results when the lexical layer was added. These measures can be improved on an individual level without a lexical layer.
4	Marujo et al., 2016	KPCentrality method of SDS is extended here for MDS. It works as a single layer as well as a waterfall approach.	The results show an improvement of 16% at ROUGE-1 scores for TAC 2009 and 17% for DUC 2007. Researchers' need to work on the area as intermediate summaries do not include all important events.
5	Lin et al., 2010	This technique provides interactive, high-quality summary generation using MMR that keeps the user in a loop during processing.	Since the user is on-board during the summarization process, it is pretty effective, but at the same time, it causes delays due to human interactions, thus, is time-consuming. Sentences with varying lengths are not handled.
6	Ozsoy et al., 2010	The Cross method is presented to handle the issue of noisy sentences selected for summary, causing the error. The topic method is used to identify the main\subtopics of sentences.	A simple and effective technique of summarization. Currently, it is tested on different scientific article of Turkish language. If tested on standard datasets, it will be helpful in the research for better comparisons.
7	Gong & Liu, 2001	SVD was used to derive the Latent Semantic structure. Weights are assigned to sentences, and those with more weights are selected in summary.	It is one of the earlier studies in this field, so it would be best to test it on benchmark datasets. Moreover, slight disparities in sentence selection are observed, which increases with the length of the documents.
8	Carbonell & Goldstein, 1998	MMR approach considers the relevance of sentences with query and other sentences in the documents.	As long as the topic remains the same, the results are promising. It doesn't work well to extract sentences from multiple topics in documents and offers a fertile field for a researcher for improvements.

2.1.2.3 Non-Negative Matrix Factorization (NMF) Methods

In non-negative matrix factorization-based methods, factorization is performed on the sentence-term matrix to determine the highest probability sentences within each topic. It is more like a clustering technique with all its benefits [36], [37], [68]. Sentences are clustered as per their set criteria, and salient sentences within clusters are then determined and summed up to create the summary.

2.1.3 Rhetoric Structure Theory-Based (RST) Methods

Rhetoric Structure Theory, or RST based methods, as depicted in Figure 2.3, divide the text into adjacent textual units that are consecutive sentences and apply different RST rules on text units to see each unit's importance. It ranks the sentences into nuclei and satellites, where nuclei are the important sentences that need to be included in the summary, and satellites contain additional information about nuclei. RST based methods are also considered in MDS [42].

Automatic Summary generation might result in poor grammatical quality. This problem is dealt with in work by Durrett et al. [42] in which Anaphoricity constraints are considered while compressing the sentences for summarization. It divides the text into text units, performs compression by Rhetoric Structure Theory by further dividing the sentence into Elementary Discourse Units (EDU), and Syntactic Compression is then applied so that the given sentence is easily compressed by considering the noun phrases, pronoun phrases, and other RST based rules while selecting the EDUs like elaboration statements for deletion. It also uses the pronoun replacement to remove any ambiguity and inconsistency from the summary. A situation arises when the statement with a pronoun is included in the summary while its antecedent (the statement containing the actual proper noun or simply the noun) is omitted from inclusion. The system then replaces the pronoun in two possible ways. It picks the noun from the antecedent statement. It replaces it with the pronoun used in the selected statement, or in case the replacement is not that straightforward, it includes the entire antecedent statement in summary. Supervised learning is done through the structured SVM technique. This algorithm, however, worked for single-document summarization.

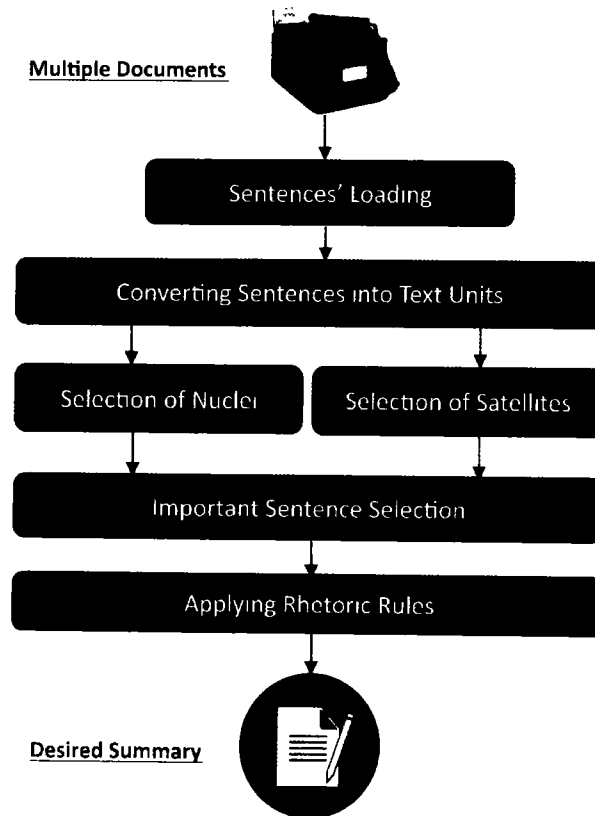


Figure 2.3: RST-Based multi-document summarization

2.1.4 Graph-Based Methods

As presented in Figure 4, graph-based methods construct graphs of sentences that are part of the document collection. The sentences make the graph's nodes, and edges are either drawn based on the similarity between sentences fulfilling the threshold criteria or belongingness to the same document. Voting of neighboring nodes selects sentences to generate a summary. Erkan and Radev [30] devised the LexRank algorithm based on eigenvector centrality (prestige) to determine significant sentences, as was done successfully in the Google PageRank algorithm.

Canhasi [69] presented a technique based on Five-Layered Heterogeneous Graph and Universal Paraphrastic Embeddings for query-focused EMDS. In this work, the focus is on sentence and document level relations and includes part of sentence similarity and query to sentence similarity.

Sentences are iteratively ranked using the PageRank [30] algorithm. To calculate the text similarity, universal paraphrase embeddings are used. The technique in this paper was implemented on benchmark dataset DUC 2005. The performance was evaluated on ROUGE 1

and ROUGE 2 as next to reference summary, while on ROUGE SU4, their performance deteriorated.

Shafiei et al. [46] presented a word graph-based method for the multi-sentence compression (MSC) approach. They used substantial merging, mapping, and re-ranking modules that resulted in more compressed summaries by retaining informative and grammatically sound sentences. Multiword Expressions (MWE) are handled by substituting an MWE with its one-word synonym and make it a node in the graph. This removes ambiguity and results in compression as well. It handles the concept of synonymy by replacing the up-coming one-word with its already existing synonym node in the graph. It uses a 7-gram POS-based language model (POS-LM) to rank the k-shortest paths obtained from the graph without compromising the resulting compressed sentence's grammar. It can be said safely that this is the first time to use MWEs, synonymy, and POS-LM for improvement in the quality of word graph-based multi-sentence compression. This approach is tested extensively on the standard datasets and has shown effective results for compression with grammaticality.

Multi-document text summarization has also experimented with data mining techniques. Baralis et al. [48] applied Association Rule Mining of data mining to see the results of its over summarization process. They devised the GRAPHSUM algorithm to find out correlations between multiple terms in graph-based summarization. Apriori algorithm was adopted to do association rule mining to find correlation among terms, and then PageRank [29] was used to rank salient sentences.

Graph techniques are also effective in many other problem-solving methods. For instance, Chali et al. [49] presented a system for answering complex questions by the random walk method of graph-based technique and measured the effect of syntactic and semantic information in it. They measured the similarity among sentences by applying tree kernel functions in the random walk framework. By incorporating the Extended String Subsequence Kernel (ESSK) to carry out the task equivalently, they significantly extended their work.

Vertex Cover algorithm-based multi-document summarization was presented by John et al. [51] using sentences' information content. The vertex cover algorithm worked like the famous Euler's graphs. To cover all edges, a graph was constructed where vertices were a subset of the original graph. Vertices represented sentences, and edge scores represented relevance with other sentences. Vertex (which was a sentence) with a higher relevance score would appear in the final graph, i.e., summary in this case.

Archetypal analysis is an unsupervised learning technique that works in the same manner as cluster analysis. Archetypes differ from common observations like cluster centers in that they are the external points in multidimensional data. A query-focused MDS [20], [52] with weighted element graphs and hierarchies was tested for improvements using archetypal analysis.

According to Tzouridis et al. [54], summaries could be created by employing the word graph to represent connected sentences, and short paths being the summaries. They used parameterized shortest path algorithm and the large margin approach for sentence compression. This approach is superior to other multi-sentence compression approaches. They used the structured approach of learning in multiple sentence compression. Parameters are adjusted in the shortest path algorithm. Data labelling is done through a structured expectation framework. Features are used to embed the word graph and its shortest paths which consequently become the desired summaries. The linear scoring function learns to differentiate between the different quality of compressions. The integer linear program is used to solve the issue that works in polynomial time. Related sentences become input to word graphs. Unique words of sentences become vertices of the graph and directed edges that connect words of at least a sentence. A path in the graph is the connected sentence. It extends the work to a structured prediction framework using parameterized shortest path algorithm. It uses SVM for the shortest path algorithm to learn the shortest path for a highly dimensional feature and proposes a polynomial-time procedure. It also uses the shortest path for the experiment of significant news. The edge weights are used based on word frequency. Some scientists use the key phrase method to generate summaries. The words used for the graph must be pre-processed. Sometimes, complex pre-processing is required, such as reunion vertices containing replacements. The shortest path algorithm figures the cost by adding all the edges in the path, such the path p has vertices between Start and End. To summarize related sentences, it is needed to find the function that gives the best summary and assign the minimum score to the best summary. To assess function f , a hamming function is used. Then the task is to find the position function that gives the smallest score to the best summary. After that margin-rescaling technique is used. The margin method is used to fetch the margin among the best path and all other paths. Decoding of P^\wedge is used for margin scaling to scale the margin with the real loss. The margin technique also affects the central loss, which is greater than structural loss. Experiments were done on a set of a predefined set of categories i.e., news about sports business, etc., and the pre-processing was done by using spectral clustering. A fully connected graph was created in which vertices were

headlines, edges were weighted by the number of shared non-stopwords. After that clustering was achieved day by day, and the resulting data was considered as headline news about the event. The data with high probability was measured as the data about the occurrence, and it was the related input sentences. For best summary identification, crowdsourcing is used. The annotator has given n number of sentences and must create ten summaries using Yen's algorithm. After that, the best summary is marked. Then three most appropriate summaries are collected. The learning approach uses the following method: Every edge is associated with a feature vector. The feature vector consists of the join frequency, maximal word frequency, lexical relevance, normalized Pointwise Mutual Information (PMI), the average location of the phrase. The experiment uses the holdout method with a distinct holdout method and test sets. Analysis demonstrates a connection between the negative correlation of lexical diversity and the positive correlation of graph density.

TH-27408

The method developed in [56] uses a graph-based approach, where nodes are represented by sentences and edges describe the sentence's preference value. It employs the entailment technique, in which the meaning of one statement can be achieved by the meaning of another one. This entailment can be found by some symmetric and non-symmetric measures. In pre-processing unit, tokenization is performed, and stop words are removed. The significance of the word in the similarity matrix is calculated through the tf-idf and weight. After that, sentence ordering is achieved based on preference measures that comprise topical closeness, chronology, precedence, succedence, semantic, and text entailment experts. This system deals with the semantic relationship, rational conclusion so that the meaningful summary is generated and emphasizes evidence extraction and sentence order. WordNet is used for the semantic link between the sentences, which creates the rational entailment between the summary sentences. The primary module is text entailment expert that investigates the logical relationship among the sentences by using symmetric and non-symmetric measures. The symmetric measure is calculated by using a cosine measure to find the similarity statistically. The additional module is the ordering of sentences. The sentences are extracted from the documents, and the total preference value is calculated. After that, the ordering algorithm will perform the ordering in the following way. Experimental results reveal that the entailment method of sentence ordering and ranking provides high precision and provides a well-organized summary that significantly helps the reader realize data. This technique, however, does not focus on coherent summaries. Similarly, the use of non-symmetric similarity measures and complex algorithms make it a bit costly solution.

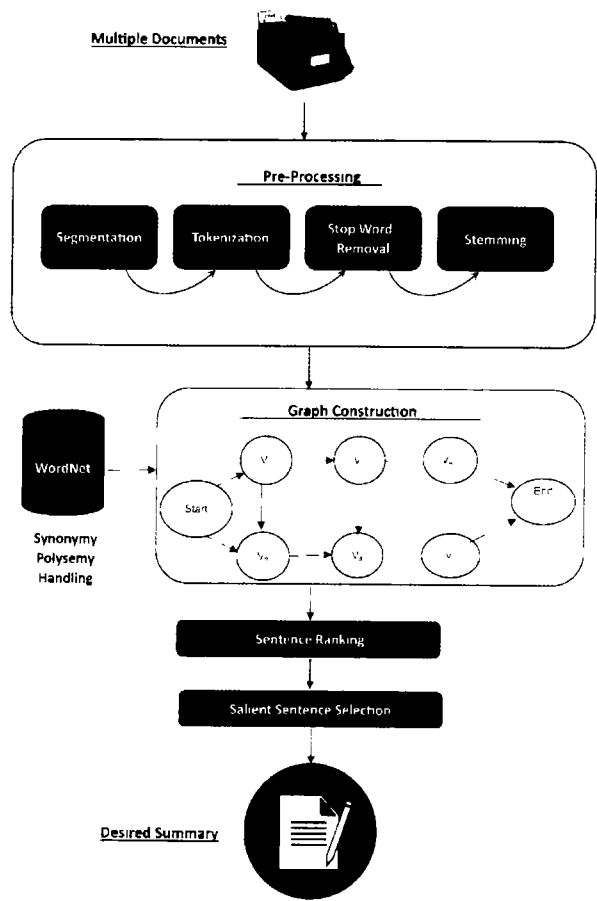


Figure 2.4: Graph-based multi-document summarization

The technique in [65] focused on G-FLOW is a novel method using the joint model. The work focuses on the technique used to resolve the problem of selecting sentences along with the sentence ordering problem. It constructs the directed graph, where sentence represents vertex and connection between the sentences s_i and s_j means that s_j can be used immediately following s_i in summary. Need is to identify sentences that have the relationship among them. This method first automatically constructs the graph for multi-document summarization, which requires innovative methods for identifying inter-document connections. It then uses this graph to find the coherence of the specific sentence. After that, G-FLOW uses a technique for sentence collection and order. Previous procedures did not emphasize coherence between sentences and selected disconnected sentences. This technique generates summaries without any domain-specific knowledge and identifies coherent documents rather than sentences. The aim is to develop a pair-wise ordering constraint which specifies a discourse graph, and which is then used by the G-FLOW graph to estimate coherence. Textual cues are from literature, and the redundancy naturally presents in connected documents used to produce edges. The technique focuses on generating coherent summaries based on jointly improving coherence and

salience. It generates a summary using ADG (approximate discourse graph) where each vertex is the sentence and edges show the discourse relationship. Experimentation demonstrations give better results than other MDS techniques. The issue is coherence and salience are less focused. WordNet is used, so more training is required.

In the graph-based methods. The technique by John and Wilsy [51] gives best results on DUC 2002 with Rouge-2 values of 0.07059, whereas in DUC 2007, Chali et al. [49] came up with Precision value of 0.392012 in Rouge-1

Table 2.4 presents the gist of methods working on the graph-based technique.

Table 2.4: Strengths and Weaknesses of Graph-Based Methods

Sr.#	Research Study	Working	Results & Evaluation
1	Davoodijam et al. (2021)	The multi-layer graph simultaneously covers semantic relationships, word relationships, and co-reference relationships	Use of UMLS ontology made the processing heavy and domain specific.
2	Tomer and Kumar et al. (2021)	Bio-inspired firefly metaheuristic algorithm to generate a summary with highly relevant sentences concerning the given topic	Coverage issue, as less frequent sentences can also be important for summary.
3	El-Kassas et al. (2020)	EdgeSumm: a mixed approach combining graphical, statistical, semantic, and centrality-based methods	Single document; pronoun resolution not handled
4	Pontes et al. (2019)	Multiple sentences are compressed using MSC, single sentences using a NN model to facilitate the translated summaries from French-to-English	Google translator was used for translation.
5	Canhasi, 2017	5-layered heterogeneous graph method is presented that also handles paraphrases.	It outperforms the other baseline implementations. However, the performance is not up-to-the-mark in ROUGE-SU4.
6	ShafieiBavani et al., 2016	A language model has been used in a word-graph-based sentence compression technique that replaces the MWEs with its one-word substitute along-with the contemporary synonym replacement.	Grammaticality is increased. It would be better to focus on the informativity of the selected summary.
7	John & Wilsy, 2015	Vertex cover algorithm is presented for text summarization that must cover all the edges in the graph. Relevance of edge defines sentence salience for inclusion in prospected summary.	Overall results are promising. Sentence relevance of the selected ones for summary needs improvement.
8	Sukumar & Gayathri, 2014	Semantic relationship and rationale are the main focus of this study by emphasizing evidence extraction and sentence ordering using the sentence entailment method.	It gives better results in sentence extraction and ordering. However, it does not focus on the coherence of summaries.
9	Tzouridis et al., 2014	Word-graph-based compression among sentences is used with supervised learning using SVM.	Word graph technique and SVM-based learning make the compression better.

			However, a limited feature set is used, which can be tested with an enhanced feature set.
10	Christensen et al., 2013	Deverbal noun method is presented for sentence selection and order problems.	With increased training, better results are expected.
11	Baralis, Cagliero, Mahoto, et al., 2013	Apriori algorithm of association rules is used with a graph-based technique, named as GraphSum, to find a correlation between terms.	This technique employs the Apriori algorithm, which scans the dataset many times. It is considered to stay in primary storage mostly and is an expensive solution based on time and space complexity, so efforts are needed to improve the efficiency.
12	Chali et al., 2011	The random walk method of the graph is used for the complex question answering system	Repeating entities, in summary, are not considered for dereferencing.

2.1.5 Miscellaneous Methods

The term-based multi-document summarization fails to handle synonymy and polysemy problems, while ontology-based summarization can work well only where the ontologies are already defined. The definition of ontology involves a great deal of workforce to define it. To overcome both the issues, Qiang et al. [15] came up with a closed pattern-based technique for MDS, which extracts the important sentences from document collection using closed patterns to decrease repetition in summary. Their method, PatSum, calculates the sentence weight in the document(s) by adding the weights of its covering closed patterns concerning the current sentence and repeatedly selecting a sentence with least closeness to the previously chosen sentences and highest weight, till the length limit is reached. This technique reduces the dimension while retaining the related information. PatSum method uses the advantages offered by the term-based and ontology-based methods without adopting their weaknesses. Extensive experiments on the benchmark DUC2004 datasets show that the pattern-based method outperforms the state-of-the-art methods significantly.

The search space is optimized using evolutionary techniques. The Cuckoo Search (CS) technique is used in work by Rautray and Balabantaray [70] to address the generic EMDS problem. The Particle Swarm Optimization (PSO) and Cat Swarm Optimization based (CSO) summarizers are two alternative evolutionary algorithms that the authors have compared to their approach. They have found out that CS-based summarizer results are better on the benchmark datasets of DUC 2006 and DUC 2007. However, since CS belongs to the evolutionary algorithms, they have an issue with controlling parameters. Therefore, this was also faced in the implementation of CS in generating summaries in MDS.

Bat Algorithm of optimization [72] was used to the objective function in search of the optimal solution. At the start, the data is divided into sentences, which consequently are divided into words. Then pre-processing is applied by removing stop words and converting the data into lower case. The objective function is designed to address two objectives:

- a) It should give proper coverage
- b) Redundancy should be avoided in summary sentences

Indian dataset is used to test the technique. Indian dataset contains 4516 news articles along-with the gold standard summaries. For the evaluation, ROGUE 1, ROGUE 2 are used, and the comparison of the summary of their technique was made with the summary generated by MS Word.

The three most essential points the best summary must contain are coverage, non-redundancy, and relevance. To achieve such a summary, the authors [73] used Shark Smell Optimization (SSO) for MDS. SSO uses the word embedding-based similarity function and Google-based similarity function, and SSO calculates optimal weights of text features. Word Mover's Distance is a word embedding technique-based distance function to find the similarity among the text documents so that the embedded words of the first document need to travel to reach the embedded words of the second document. In contrast, Normalized Google Distance is a Google hit-based dissimilarity function.

The technique was tested on DUC 2004, DUC 2006, DUC 2007, TAC 2008, TAC 2011, and MultiLing 13.

In this paper [75], the authors devised three methods for sentence selection, namely sentence-context relevance, sentence novelty, and sentence position relevance for the methodology SummCoder for a summary generation. These sentence features are fused to rank and select sentences for a summary of the given length. TIDSum dataset is used to test the methodology, along-with DUC 2002, and Blog Summarization Corpus. Unsupervised deep auto-encoder was trained such that Recurrent Neural Networks (RNNs) encoder with Gated Recurrent Units (GRUs) and RNN decoder with conditional GRUs.

ROGUE recall factor for R1, R2, Rogue L, ROGUE SU4 are applied.

There are different other methods like CRF-based summarization and Hidden Markov Model (HMM) based method. Table 2.5 shows the pros and cons of miscellaneous methods working in MSD.

Table 2.5: Strengths and Weaknesses of Miscellaneous Methods

Sr.#	Research Study	Working	Results & Evaluation
1	Joshi et al., 2019	SummCoder technique is proposed, which comprises sentence-context relevance, sentence novelty, and sentence position relevance.	It gives promising results on single-document summarization. Can be extended on MDS
2	Anshuman Pattanaik, Santwana Sagnika & Mishra, 2019	Bat algorithm for optimization is used to search the optimal solution, to maximize the coverage and minimize the repetition	Can be experimented on benchmark datasets like DUC, TAC, and others.
3	Verma & Om, 2019	MCRMR algorithm is designed by using the Shark Smell Optimization technique on MDS for best results	With Machine Learning based methods, the results can become better.
4	Naserasadi, A., Khosravi, H., & Sadeghi, F. (2019).	Sentences are ranked, and then entailment scores are calculated and then finally compressed using 0-1 Knapsack problem.	Gives 2% improvement in the query-based approach, while 5% improvement is recorded for the generic summary. Efforts are required to decrease the complexity of the algorithm
5	Rautray & Balabantaray, 2018	An Evolutionary algorithm, called Cuckoo Search, is applied in MDS.	The parameter controlling problem of evolutionary algorithms needs to be resolved.
6	Qiang et al., 2016	Closed patterns are applied to find the shortest path for MDS. The solution, named PatSum, is compared with ontology and term-based methods.	For larger support value, the performance of PatSum declines.

2.1.6 Secondary Studies Conducted in MDS

MDS has attracted many authors for performing secondary studies as well.

This paper [76] briefly discusses the different techniques of extractive and abstractive summarization. They explored the different pros and cons of both types of summarizations and proposed that a mixed approach should be used for better summary generation.

A detailed survey is conducted to investigate the focus of current studies in text summarization [77]. The authors also helped the new researchers by projecting the research gap in this field. A similar survey was conducted by [78] on legal documents. The study examined the text summarizing strategies created for the summarization of legal documents and gathered performance comparisons of various approaches and datasets for the interested scholars.

In another secondary study [79], a systematic literature review was conducted to investigate the status of importance and significance of fuzzy logic in text summarization. They designed the research questions to conduct this study on electronic research databases, like, IEEEExplore, ACM Digital Library, ScienceDirect, GoogleScholar, Springer, and Wiley Digital Online. After performing the respective inclusion-exclusion, 52 articles qualified to be included in this SLR. 49 were primary studies, and 3 were secondary studies on fuzzy logic for text summarization. Further quality assessments finally resulted in 42 total studies in SLR, 39 were primary studies, and 3 were secondary studies. The findings of SLR affirmed the importance and emerging trend of the use of fuzzy logic in text summarization.

2.2 Datasets

DUC-Document Understanding Conference: Since 2001, the Document Understanding Conferences is playing the role of an effective forum for researchers in automatic text summarization to compare common test sets' methods and results. They release datasets having benchmark document collections from multiple sources on an almost yearly basis. Additionally, it contains the reference summaries created by humans so that users can compare their candidate summaries (produced by the various algorithms) with them [7], [8], [47]. A majority of authors [1] – [8], [11] – [13], [15] – [20], [24], [26], [31], [34], [40], [47]– [49], [51] – [53], [60], [71], [80], have employed DUC to track the effectiveness of their method.

TAC- Text Analysis Conference: Similar to DUC, TAC is a benchmarked compilation of documents from various sources with reference summaries created by human experts. The distinction is that TAC is expanded to include support for additional languages [10], [13], [14]. The authors [14], [17], [47], [52] tested their techniques on TAC.

The following other datasets were also used:

TSC-3 – (Text Summarization Challenge corpus) is utilized by [42], [44]. Similarly, RSS Feeds, New York Times annotated corpus, TREC 2007 are used, as well as the datasets created by users.

2.3 Evaluation Techniques

ROUGE - Recall-Oriented Understudy for Gisting Evaluation: For evaluating automatic summarization and machine translation software in NLP, it is a set of measurements and tools. The metrics evaluate a reference or group of references (human-produced) summary or translation with an autonomously generated summary or translation. Authors in [1], [3] – [8],

[10]– [20], [24], [25], [29] [35], [39], [41], [43], [48], [51], [54], [55], [62]–[68] , [70], [71], [80] used ROUGE for the evaluation of their systems.

BLEU- Bilingual Evaluation Understudy: It is a unique algorithm for assessing the quality of content that has been machine translated between natural languages. It evaluates the translation done by machine with its closeness with human translation on the measure of fluency and adequacy. It is used in [46], [54], [65].

Other evaluation metrics used are Precision, Recall, F-measure, Average Continuity, Pyramid, SemEval, Correlation Coefficients, Amazon mTurk, etc.

2.4 Research Gap and Limitations

The current literature review presents an insightful discussion on extractive MDS techniques. However, these techniques for extractive MDS have following limitations.

1. The data processing techniques for big data are difficult for users to understand and adapt as a solution for text summarization [55].
2. The centroid-based techniques are inherently complex solution for extractive MDS [10],[21] - [26],[55],[60],[62],[66].
3. The term-based techniques for extractive MDS fail to address the issue of polysemy and synonymy [15].
4. The centroid-based techniques have many problems like it is difficult to predict k value; with global cluster or with clusters of different size and density, their performance deteriorate; different initial partitions can result in difficult final clusters.
5. Besides, the LSA-based extractive MDS techniques doesn't focus on word order or syntactic relations which is required for finding out the meaning of words.
6. Moreover, RST-based techniques have strong dependency on Rhetoric Relations [42] which is a challenge for researchers.
7. The graph-based techniques for EMDS miss the Direct Similarity Analysis [3].
8. Furthermore, ontology-based techniques for extractive MDS fail to perform where ontologies are missing, while generating the missing ontologies is a process that demands a lot of manpower [15].

These challenges are presented in figure 2.5.

Problem Statement

The process of summarization, whether single document or multi document, renders in a compromised grammaticality. The situation becomes more worse when it comes to extractive summarization, as it picks the chunks of sentences from the given document(s) to present to the user the desired summary. Moreover, the resultant summaries lack informativity, diversity, and contains redundancy. Therefore, the need is to have an extractive MDS solution that produces a summary that informative, diverse, non-redundant, and grammatically correct.

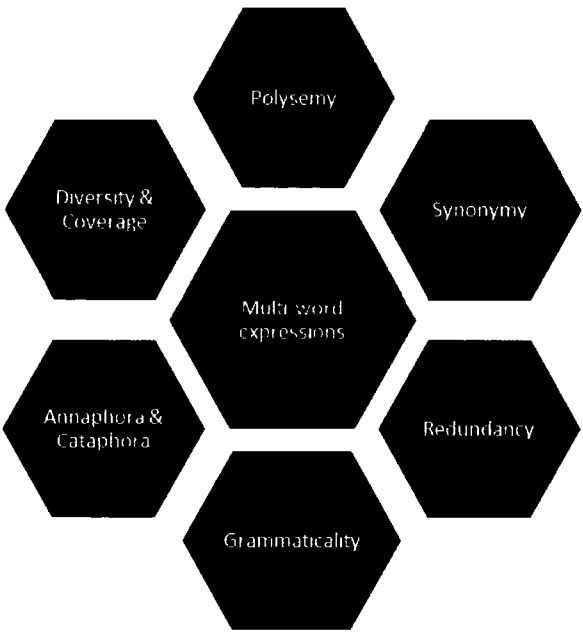


Figure 2.5: Challenges of EMDS

Proposed Solution

To devise and develop an extractive MDS framework that addresses the challenges like lack of informativity, diversity, redundancy mitigation, and grammaticality. We need to adapt different solution strategies, i.e., topic modelling, synonym mapping, pronoun replacement, multi-word expressions. In this regard, we would apply these solutions strategies to graph-based and RST-based extractive MDS techniques. Later, we can compare our results with state-of-the-art in extractive MDS techniques, term based, ontology based, close patten based extractive MDS techniques.

One of the important aspects of solution to the MDS is machine learning and advances in this field are tremendous [42,54]. If we use features like the significance of vertices, edges, sentence position in document etc., and add handcrafted rules to generate a weighted graph and

summary based on features then it can be a potential valuable summarization solution. Therefore, it is worthwhile to explore machine learning in field of extractive MDS.

Chapter 3

Research Method

3 Aim

“To explore, analyse, devise, and evaluate an Extractive Multi-Document Summarization Framework using Learning based Graph Theory and Rhetoric Structure Theory in comparison with state-of-the-art techniques w.r.t. informativity, grammaticality, diversity and redundancy mitigation on different datasets.”

3.1 Research Objectives and Contributions

Objective 1: “To explore the current research front in the area of extractive multi-document summarization.”

Contribution: A literature survey was conducted to investigate the current evidence on EMDS

Objective 2: “To devise and implement an extractive multi-document summarization system using graph theory.”

Contribution: An extractive multi-document summarization system using graph theory was devised and implemented.

Objective 3: “To devise and implement a learning based extractive multi-document summarization system using graph theory.”

Contribution: A learning based extractive multi-document summarization system using graph theory was devised and implemented.

Objective 4: “To devise and implement an extractive multi-document summarization system using rhetoric structure theory.”

Contribution: An extractive multi-document summarization system using rhetoric structure theory was devised and implemented.

Objective 5: “To implement the state-of-the-art baselines for the purpose of comparison.

5.1 to implement term-based EMDS system.

5.2 to implement YAGO ontology-based EMDS system.

5.3 to implement Closed Pattern based EMDS system.”

Contribution: Implemented the state-of-the-art baseline systems for the purpose comparison

5.1 implemented term-based EMDS system.

5.2 implemented YAGO ontology-based EMDS system.

5.3 implemented Closed Pattern based EMDS system.

Objective 6: “To perform a comparative evaluation of EMDS systems using graph based and RST based systems with state-of-the-art baseline systems, i.e., term based, ontology based, closed pattern based systems on different datasets using Qualitative and Quantitative parameters.”

Contribution: A comparative evaluation was performed between EMDS systems using graph based and RST based systems with state-of-the-art baseline systems, i.e., term based, ontology based, closed pattern based systems using ROUGE evaluation and other quality parameters i.e., informativity, coverage, grammaticality, and redundancy mitigation on different datasets.

3.2 Research Questions

Question 1: What research evidence has been reported in the literature on extractive multi-document summarization?

Answer: we have conducted a thorough literature review [81] that mentions research evidence in the field of extractive MDS.

Question 2: How to devise and implement an EMDS system using graph theory?

Answer: We have conducted an experiment (mentioned in chapter 4) that sheds the light on the improvements brought by the graph-based method in the field of extractive MDS.

Question 3: How to devise and implement learning based EMDS system using graph theory?

Answer: We have performed an experiment that mentions the improvements by the machine learning techniques to extractive MDS. We have discussed it in chapter 7.

Question 4: How to design and implement RST based EMDS system?

Answer: In chapter 5, we have discussed another experiment that we have conducted to show the improvements of RST based summarizers to the extractive multi-document summarization.

Question 5: How to implement SOTA baseline systems for the purpose of comparison?

5.1: How to implement term based EMDS system

5.2: How to implement closed pattern based EMDS system

5.3: How to implement ontology based EMDS system

Answer: Experiments conducted that are stated in chapter 6 answers this question and its sub-parts.

Question 6: Which of the following techniques perform better in comparison with the SOTA EMDS systems (i.e., term based, ontology based, close pattern based) w.r.t. qualitative (informativity, representativeness, grammaticality, diversity, redundancy mitigation etc.) and quantitative (ROUGE evaluation) parameters?

- a. Graph based EMDS system.
- b. Machine learning based EMDS system.
- c. RST based EMDS system.

Answer: Experiments conducted that are stated in chapter 6 and 7 answers this question.

3.3 Proposed Solution

To address challenges such as lack of informative content, diversity, redundancy mitigation, and grammatical accuracy, our objective is to devise and develop an EMDS framework. We aim to accomplish this by employing various solution strategies, namely topic modeling, synonym mapping, pronoun replacement, and handling multi-word expressions. These solution strategies will be applied to both graph-based and RST-based extractive MDS techniques. Subsequently, we will compare our results with state-of-the-art approaches in extractive MDS, including term-based, ontology-based, and close pattern-based techniques.

Within the context of MDS solutions, machine learning plays a crucial role, and recent advancements in this field have been remarkable. By incorporating features (discussed in subsequent section) and applying handcrafted rules to generate a weighted graph and summary based on these features, we can potentially develop a valuable summarization solution. Consequently, it is highly worthwhile to explore the application of machine learning in the field of extractive MDS. Therefore, the proposed technique is divided into 5 main modules:

3.3.1 Module 1

In module 1, the aim of the study is to investigate the graph-based extractive MDS technique to give best results that are near to human generated reference summaries. It requires a set of features to perform the job which are stated below:

Features:

Finding sentence relevance from a single point of view is challenging. The inclusion of a sentence in the summary is decided using a few features. To find the significant sentences, some machine learning-based methods that select the significant sentence are used. Surface, content (a key feature), sentences to sentences unity, discourse analysis, occurrence of unimportant information, relevance, or some event features are a few of the event features. In supervised approaches, a classifier is trained for the provided text using a collection of documents and manually created summaries. These methods make advantage of feature-based training. Features include word counts in the sentences and sentence positions that make them suitable candidates for inclusion in the summary. The use of SVM to judge the importance of a sentence uses feature categories: Surface features (location, sentence length, etc), Content features (uses statistics of content-bearing words), Relevance features (Uses inter-sentence relationship e.g. similarity of a given sentence with the first sentence). In the proposed work, the features are based on the assignment as v, v' are two vertices having an edge associating features with itself. Then $w = \#(v)$ frequency of word v ; $w' = \#(v')$ frequency of word v' ; $e = \#(v, v')$ frequency of edge between v and v' ; $n = \#$ number of vertices in graph. The features are:

1. Joint frequency $\phi_1(w, w') = e/n = \text{frequency of edge/no. of vertices in graph}$.
2. Maximal word frequency $\phi_2(w, w') = \max \{w/n, w'/n\} = \max \{\text{freq. of } v/\text{no. of vertices, freq. of } v'/\text{no. of vertices}\}$
3. Lexical relevance $\phi_3(w, w') = (2/n) \cdot (w \cdot w' / (w + w'))$
4. Normalized PMI $\phi_4(w, w') = (\log(e/w \cdot w')) / (-\log e/n) = (\log e/w \cdot w') / -\text{joint frequency}$
5. Average location of the phrase or group of words in the input sentence

Along with these features, additional features are included like:

Lexical: Indicator features on non-stopwords that appear more than 5 times in training set and its similar part-of-speech features. First word, last word, preceding word and following words in each textual unit are to be considered, along with the index of sentence having the textual unit in the document.

Structural: It includes the conjunctions of position of textual unit in the document, its length, length of the corresponding sentence, index of the paragraph it occurs in, whether it is start of the new paragraph.

Centrality: It's about centrality of content, word counts coupled with sentence index in multiple documents. Also, to include in features that how much each entity mentioned in the sentence is mentioned versus the rest of the documents, the mentioned no. of entities in the sentence, surface properties like the type and length of mentions.

Pronoun replacement: Such as the length of pronoun replacement, its sentence distance from the current mention, its type (nominal or proper), identity of the pronoun being replaced.

Using these features, we will devise two techniques. First technique is graph based and second technique is RST based.

The Graph based Extractive MDS

It is a well-known fact that graph-based methods are easier for the computer users to understand as graphs are widely used in its core problem solving techniques in algorithms [48]. Therefore, a study was conducted to find out a novel graph-based method to facilitate the job of MDS, exhibiting machine learning behavior. It extracts the sentences from multiple documents and tokenize them. In the process of tokenization, the system will add START and END token to every sentence. Similarly, POS-Tagging, Stemming, Lemmatization, and Stop-Word removal are also performed in this initial pre-processing stage. Word graph, as adapted from Filippova [82] was constructed by loading each sentence in a word-by-word fashion to construct Word Graph. Unique words become the vertices of the graph and its connection as edges. Since a word is followed by another, therefore it is a directed graph to keep the order of the sentence intact. The word once selected as a vertex in the graph if encountered again, is redirected to the existing vertex, by adding a new edge towards the vertex, hence eliminating the duplicate vertices. There is a possibility that two vertices are created with same meanings, as it is one of the old problems of synonymy. In order to deal with that, we took advantage of Thesaurus.com, and it assigns the same vertex to the synonyms. The idea is adapted from the work of Tzouridis et al. [54] who applied it for sentence compression, while we applied it on MDS. It is important to note that since the input for word graph construction are sentences, which may contain Multi-Word Expressions (MWE). MWE are the phrases that can be explained in a single word. Example of MWE is presented in Figure 3.1 where the MWE “kick the bucket” is mapped with its single word substitute “die” [46].

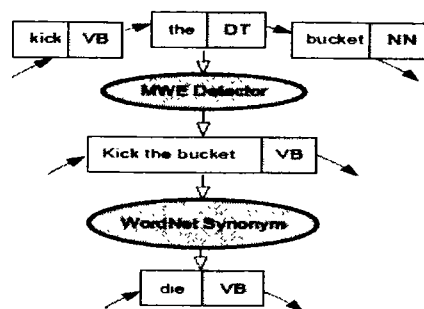


Figure 3.1: Example of Multi-Word Expression mapping [46]

The input documents always represent different nouns with pronouns once it is mentioned. Therefore, pronoun replacement will also be handled by the anaphora [42] and cataphora used in sentences that qualify for summarization.

Anaphora and Cataphora are defined as: “anaphora is the use of an expression whose interpretation depends upon another expression in context (its antecedent or postcedent). In a narrower sense, anaphora is the use of an expression that depends specifically upon an antecedent expression and thus is contrasted with cataphora, which is the use of an expression that depends upon a postcedent expression. The anaphoric (referring) term is called an **anaphor**. For example, in the sentence *Sally arrived, but nobody saw her*, the pronoun *her* is an anaphor, referring back to the antecedent *Sally*. Cataphora is the use of an expression or word that co-refers with a later, more specific, expression in the discourse. The preceding expression, whose meaning is determined or specified by the later expression, may be called a **cataphor**. Examples are:

- If you want **some**, here's some **parmesan cheese**.
- After **he** had received his orders, **the soldier** left the barracks.”

Cost Function assigns positive weights to every edge of the graph. The then constructed word graph is then checked to find out the shortest path in the graph starting from the START vertex and ending at the END vertex. The path with the minimum cost will be the shortest path of the graph. It is also agreed to add machine learning capabilities to the system; therefore, the system need K best paths in the graph for training. Yen’s algorithm will be used to do that. These k-shortest paths will be given as input to SVM for machine learning. As depicted in figure 3.2, the SVM applies ranking function to find second best, third best, and so on up to k-best paths in the graph. Hamming loss will be used to calculate the difference between the expected path and the best path. Optimization and margin rescaling will be used to facilitate the placing of hyperplane at best possible place for classification. The paths will be the desired summary from the multiple documents.

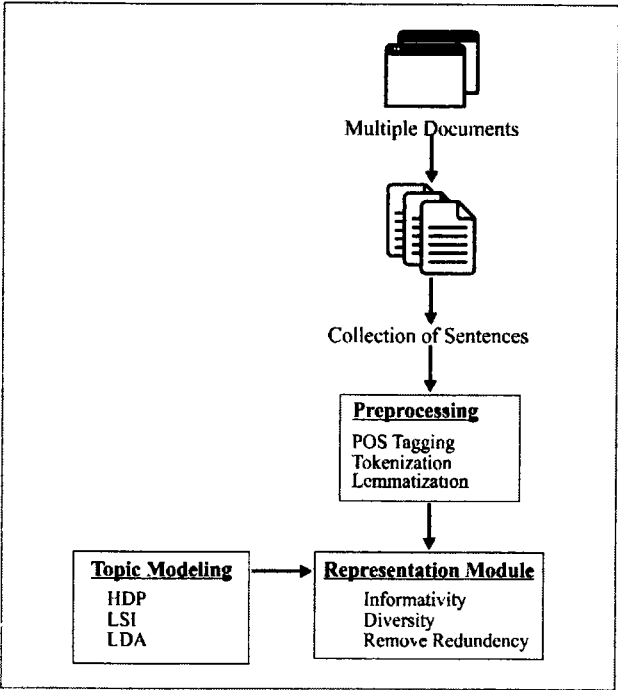


Figure 3.2: Module 1- Graph based Extractive MDS: pre-processing

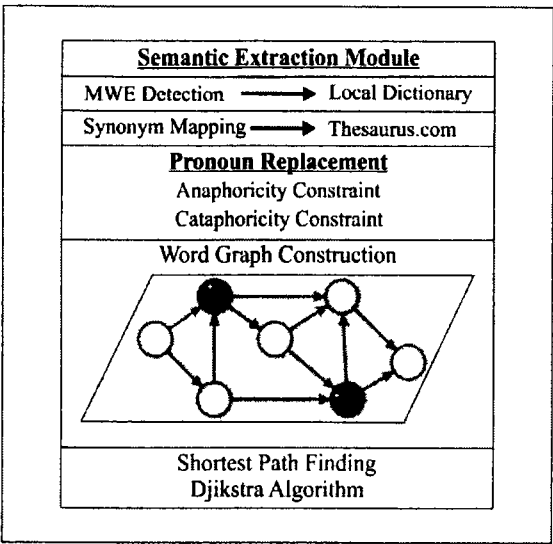


Figure 3.3: Module 1- Graph based Extractive MDS: The processing.

3.3.2 Module 2- The RST based Extractive MDS

The RST based technique is important to extract compressed summaries, as it divides the documents into textual units first. Then it applies different rhetorical relations over those units to decide about the important parts of the sentences for inclusion in the summary, just as mentioned by Durrett et al. [42]. For example, it takes a sentence as a textual unit, and further divides it into elementary discourse units EDU. It then investigates the nature of different EDUs of the sentence that whether they are same units or elaboration of the previous/ next unit by employing the rhetorical relations. It also assigns the units the different parts of speech annotations. Hence it becomes easy to identify the relation between different EDUs which in turn makes the inclusion/exclusion of the EDU trivial. It also takes care of the pronoun replacement. Therefore, it checks for the pronoun used in a sentence selected to be include in the summary. The sentence in which the pronoun's actual noun is mentioned is named as antecedent sentence of the said sentence containing the pronoun. This method of pronoun replacement will now have to calculate for the actual noun of the pronoun mention. If it is straight forward mention, it simply replaces the pronoun with the actual noun from the antecedent sentence. Else, if the pronoun mention in antecedent is not straight forward, then it includes the entire antecedent sentence into the summary as well for the clarity's sake. Figure 3.4 shows this functionality in detail.

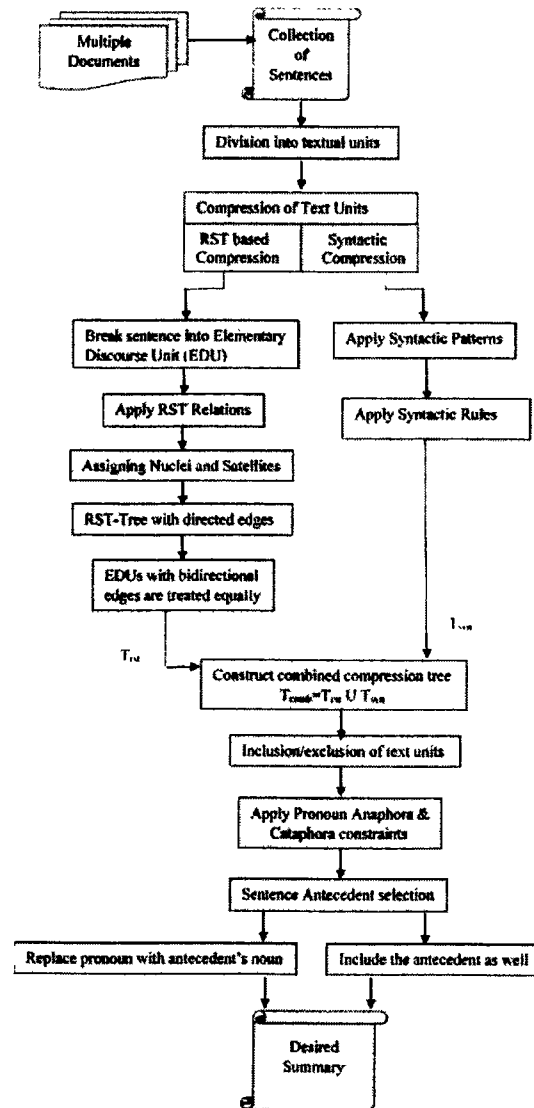


Figure 3.4: Module 2- RST based Extractive MDS

3.3.3 Module 3: Machine Learning for EMDS

In the module 3, the aim of the study is to gauge the effects of machine learning over the techniques discussed in section 3.3.1. The technique is provided with training data and then is tested for the machine learning. The details are given in the following sections.

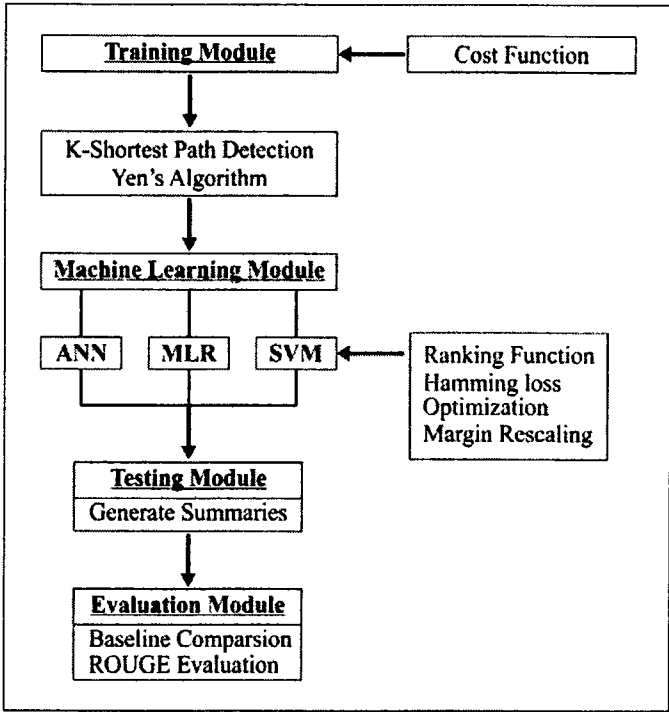


Figure 3.5: Module 1- Graph based Extractive MDS: Machine Learning

Effects of Machine Learning via SVM on graph-based Extractive MDS

Support Vector Machine or SVM is a well-known supervised learning classifier used extensively in the field of research and development. The application of SVM over technique discussed in section 3.3.1 is described in the figure 3.4. For the technique discussed in section 3.3.2, the details are in figure 3.5. SVM uses quadratic programming for optimization. It draws hyperplanes with different weights and tries to maximize the gap between given classes. The selection of a hyperplane among the others is since it should give optimized solution which needs optimization based on by maximizing the gap between support vectors on basis of some constraints. Ranking is performed on training data like when we search some string in google, it provides the most relevant ones at the top based on the rank SVM. It also uses loss + penalty paradigm for optimization by modifying quadratic optimization problem thus enabling the use of simple quadratic optimization as well as loss + penalty form. SVM uses hinge loss. If the data is nonlinear, it either performs margin rescaling or slack rescaling (which is the penalty) [42],[54].

Effects of Machine Learning via ANN on graph-based Extractive MDS

Artificial Neural Networks, or simply ANN is a popular learning technique that is inspired by the working of neurons in the human body. An ANN is a paradigm for information processing that draws inspiration from how biological nervous systems, like the brain, process information. The innovative structure of the information processing system is the fundamental component of this paradigm. It is made up of several, highly interconnected processing units (neurons) that collaborate to address particular issues. ANNs learn via imitation just like people do. Through a learning process, an ANN is tailored for a particular purpose, such as pattern recognition or data classification. The synaptic connections between the neurons in biological systems change as a result of learning. This also applies to ANNs. Wang et al. [83] incorporated it in abstractive text summarization.

Effects of Machine Learning via MLR on graph-based Extractive MDS

One of the most used data analysis methodologies is regression analysis. A supervised machine learning algorithm is used. A helpful statistical method for analyzing the relationship between two or more variables in a dataset is regression analysis. Multivariate Regression is a supervised ML approach that analyses various data variables with one dependent variable and several independent variables. We try to anticipate the outcome based on the number of independent variables. Multivariate regression's ability to help grasp the connections between the variables in a dataset is its most helpful feature. This will make it simpler to understand how dependent and independent variables are related. Multivariate linear regression is a common machine learning method. Multivariate regression is utilized when there are multiple independent variables and ordinary linear regression is ineffective. Real-world data includes a number of variables or properties, therefore Multivariate regression is required for better analysis.

3.3.4 Module 4- Baselines comparison

In the 4th module of the research thesis, the framework is evaluated by comparing the results of the developed techniques with other techniques to see the best performance in extractive multi-document summarization. Some state-of-the-art techniques selected for comparison are Term based technique, Ontology based technique, and finally Pattern based technique, as adapted by Qiang et al. [15]. This detailed comparison can be found in chapter 6.

3.4 Research Methodology

The suitable research method for this research study is survey, followed by multi-experimental studies. Since we are not to perform this test in real life, instead we will be checking this entire framework under a controlled environment, where some of the variables are controlled variables, some independent variables whose changing values can help us to see the effects in dependent variables in the laboratory settings. Complete research strategy is given in figure 3.6. The research strategy is as discussed below.

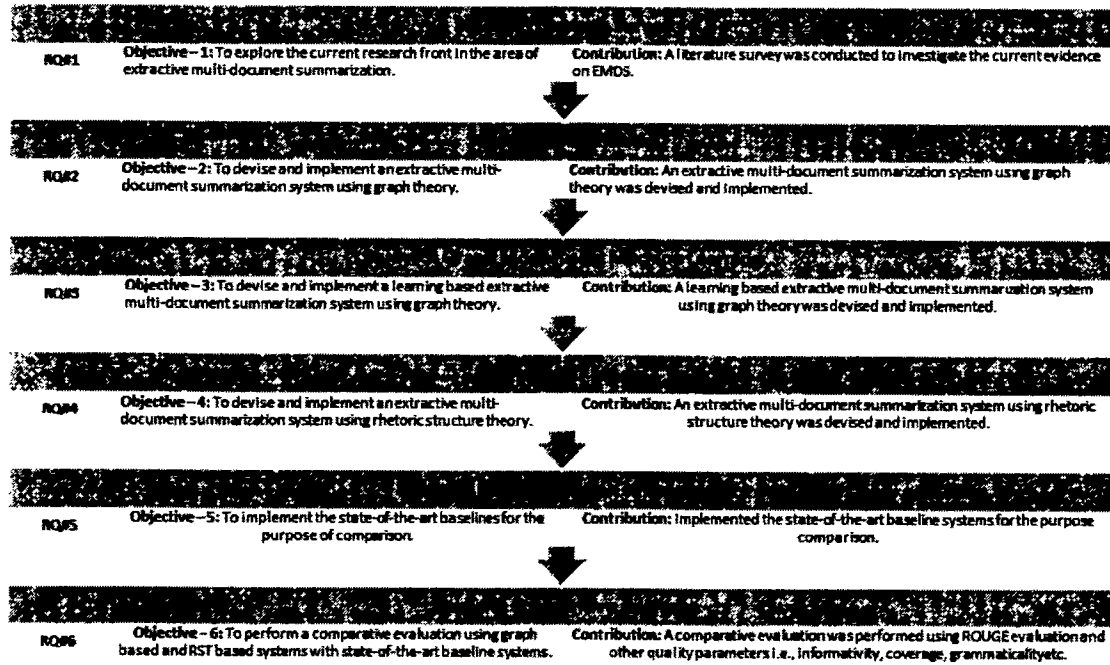


Figure 3.6: Research Strategy of Proposed Framework

3.4.1 Definition

We can define the experiment as below:

Analyze the Learning based Framework for Extractive Multi-Document Summarization to evaluate with respect to the informativity, diversity, representativeness, redundancy mitigation, and grammaticality, from the point of view of the researcher using the dataset i.e., DUC 2004 [7], [8], [47], [84] as benchmark.

Details of datasets are mentioned in chapter 2.

Object of Study:

The object of the study is a complete framework of extractive MDS, where the effect of pre-processing and machine learning will be checked, and devised systems will be compared with state-of-the-art systems available.

Purpose:

The purpose of this study is to evaluate the effect of pre-processing, i.e., synonym mapping, pronoun replacement, multi-word expression, and machine learning on EMDS using Graph-based and Rhetoric Structured based theory using DUC 2004, and recent news articles in comparison with state-of-the-art MDS systems.

Quality focus:

The quality foci (focuses) of this experiment are mentioned below:

i) Representativeness

Representativeness is used to show the importance of sentence in document set [53]. A sentence having similarity with more sentences is considered important and hence representative. Through representativeness, the less important sentences are eliminated from entering the summary. Representativeness to be used here is based on boolean topic co-occurrence. Representativeness focuses on a sentence's similarity to every other sentence in the dataset and remove any sentences that aren't crucial by subtracting a predetermined density threshold value. This is applied at topic modelling step in implementation of the system.

$$Repres(i) = \frac{1}{N} \sum_{j=1, j \neq i}^N X(sim(i, j)_{sxt} - \delta)$$

where

$$X(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{otherwise} \end{cases}$$

(1)

In equation 1, Repres(i) shows the Representativeness of the particular sentence i, whereas the sim(i,j) denotes the similarity value between ith and jth sentence and N shows the total number of sentences in the dataset. The sxt is used to measure the sentence for representativeness in the topic. The symbol of δ denotes the predefined threshold value which is used to exclude the sentences holding lower similarity with the current sentence. X is the matrix that is created after subtracting the similarity value with the specific threshold value with 0 and 1 for replacement as Boolean weighting scheme is used.

ii) Diversity

The summary should not only be focusing on a single, or few topics of the given documents, but it should instead be covering all the important topics discussed in the provided documents. Cosine diversity, as discussed in [53] selected for this study is:

$$Div_s(i) = 1 - \max_{j: Repres(j) > Repres(i)} (sim(i, j)_{sim}) \quad (2)$$

and sentences with higher representativeness are to be dealt with

$$Div_s(i) = 1 - \min_{j: i \neq j} (sim(i, j)_{sim}) \quad (3)$$

In the equation 2 and 3, $Div_s(i)$ shows the diversity value of the particular sentence present in the dataset. The diversity of a sentence is measured by calculating the minimum distance between the sentence and other sentences that have highest density. In this equation $sim(i, j)$ denotes the similarity value between i^{th} and j^{th} sentence, $Repres(i)$ shows the Representativeness of the particular sentence and max and min are used for maximum representativeness values and minimum representativeness value of the sentence. This measure is also calculated at topic modelling phase of system implementation.

iii) Length of the desired summary

In document summarizing, the objective is to draw out the most crucial information from a collection of documents. The length of the provided text should be the primary consideration while summarizing a document. For this purpose, the module named as Length is used which is based on the comparison of effective and real length [53].

$$len_s(i) = \frac{el(s_i)}{\max_{i=1}^N el(s_j)} \times \log \frac{(\max_{i=1}^N rl(s_j))}{rl(s_i)} \quad (4)$$

In this equation, the $len_s(i)$ shows the length of the particular sentence, el denotes the effective length of the sentence which means the count of distinct words in the sentence. rl shows the real length of the sentence which means the total number of word count present in the particular sentence. Max is used to find out the maximum effective and real length of the sentences. This measure is also calculated at topic modelling phase of system implementation.

iv) Redundancy mitigation

In extractive summarization, sometimes the same topic from multiple documents appears again and again hence killing the beauty of summary. This redundancy of same topic will be tackled.

v) **Grammaticality**

The automatically generated summary suffer the grammatical weaknesses. This is one of the quality foci in this study to handle it within the scope mentioned in section 3.3.

Perspective:

This research is conducted with the perspective of Researcher.

Context:

The context of this research is as follows:

Subject: Researcher, that provided different treatments to the identified problem

Object: Dataset of DUC 2004 and recent news articles. DUC is Document Understanding Conference that annually releases dataset along with human written reference summaries, which are to be compared with the automated generated summaries for evaluation. Text Analysis Conference TAC also works on the same line of DUC while we have generated the recent news articles dataset on the same format for this experiment.

3.4.2 Planning

In the planning stage, we must do the following:

Hypotheses formulation, variables selection, subject selection, design selection, instrumentation, validation. These are explained as follows.

Hypotheses formulation:

The hypotheses of this experiment are:

H₀: Machine learning and pre-processing do not improve the process of EMDS w.r.t. DUC 2004/recent news articles against any of the following systems:

1. Graph based system.
2. RST based system.
3. Ontology based system.
4. Term based system.
5. Pattern based system.

H₁: Supervised learning and pre-processing improve the process of EMDS w.r.t. DUC 2004/recent news articles against any of the following systems:

1. Graph based system.
2. RST based system.
3. Ontology based system.

4. Term based system.
5. Pattern based system.

Variable and Subject Selection:

The control independent variable of this experiment is dataset. Manipulating independent variable are the different systems that we applied. The dependent variables on which the effects are examined are informativity, diversity, representativeness, redundancy, and grammaticality. The subject of this experiment is researcher.

Treatments:

- Graph based system with machine learning.
- Graph based system without machine learning.
- RST system.
- Ontology based system.
- Term based system.
- Pattern based system.

Instrumentation:

In this experiment, we implemented the modules discussed in section 3.3. The environment in which the experiment is implemented is Python.

Validation:

A collection of metrics and software program called ROUGE, or Recall-Oriented Understudy for Gisting Evaluation, are used to assess automatic summarization and machine translation software in NLP. The metrics contrast a reference or group of references (human-produced) summary or translation with an autonomously generated summary or translation. Validation is done using ROUGE [7], [8], [10], [13], [14], [47].

3.4.3 Baselines

The proposed framework is tested against numerous contemporary algorithms. It includes ontology-based system, term-based system and pattern-based system [15], to see the improvement in the process of summarization.

Since the module 1 of the proposed work is mainly adapted from the work of Tzouridis et al. [54], therefore it is best to check it against the said work as well. We implemented it with, as well as without, the machine learning aspect to verify our hypothesis.

Similarly, the module 2 of the proposed work is influenced by the work of Durrett et al [42], therefore we tested the proposed framework with their work as well.

3.4.4 Dataset: DUC-Document Understanding Conference:

Since 2001, researchers in ATS have compared techniques and outcomes using standard test sets at the Document Understanding Conferences. They release datasets having benchmark documents collections from multiple sources on almost yearly basis. Additionally, it contains the reference summaries created by humans so that users can compare their candidate summaries (produced by the various algorithms) with them [7], [8], [47], [84].

We have also generated another dataset using the same principles of DUC 2004 to validate the results of the systems, and named it recent news articles.

Chapter 4

Graph Model

4 Introduction

With automation in every domain, the reliance of people on computers has increased tremendously. As a result, there is an exponential increase in the number of online documents as well. Bidoki et al reported that each day, about 4 million blog posts are published on the Internet which is a huge addition to the existing body on the Internet [85]. It becomes increasingly difficult to find out important information from such a huge body of online raw data. ATS has been in use for the last six decades to handle this issue. The Internet users become frustrated to find redundant information, instead of having new updates to the topic of their interest when they search. Text summarization tools can give coverage to such needs and reduce redundancy [86].

Text summarization techniques can produce a variety of summaries. Based on the nature of the technique to shrink the documents, the summary can be abstractive or extractive. The former is involved in generating a summary, after reading the document(s), using the words that are not necessarily present in the document(s), while the latter deals with the selection of important sentences from the document(s) and combines the extracts to form the summary [87]. Similarly, the input for summarization task can be a single document, making it Single-Document Summarization, or several documents, making it a Multi-Document Summarization [88]. Similarly, if training data is needed so that model is trained to learn about important sentences in documents to prepare a summary, it is the Supervised learning technique of text summarization, while in unsupervised one, there is no need for human-generated summaries, and the model needs no training data. Based on how the user needs a summary to be generated, it can be done in two ways. Either you want to generate a summary from the document(s) on a particular topic or want to know the gist of the entire document cluster(s). The former makes query-based summarization while the latter is a generic technique of summarization [88].

Extractive MDS is one of the most popular methods used in an automatic summary generation [81, 86, 89, 90]. Extractive MDS aims to mark important sentences in the document cluster and create the summary in such a manner that the repetition in the different documents must not result in the duplication of information in the summary. The summary should also, not miss the important facts found in the multiple documents while handling the issue of redundancy [85].

There are several challenges that Extractive MDS suffers from. One such challenge is the poor grammaticality of the resultant summary [42, 46]. Lots of the research is focused nowadays to

improve the sentence quality of the summary. Since the input documents are containing similar content in different words and forms, therefore it is important to pre-process such similar words or phrases from input data that are mentioned differently through effective synonym mapping. It is also found that some articles contain phrases to express some situation or point. Such phrases are known as MWE [46]. If all the MWEs are substituted with appropriate single-word synonyms, the resultant concise summary will offer a better grammatical ground.

Grammaticality suffers the most in anaphora and cataphora. It is not necessary that the sentences containing the nouns are picked up for summary. On the contrary, if the sentences with pronouns are included in the summary without their original noun mentions, the summary will become meaningless. Therefore, the pronoun replacement must be carefully done to keep the summary grammatically correct [42, 87].

The query-focused summarization process is straightforward, i.e., the model shall pick all the sentences containing the keywords of the query to prepare the summary [90]. The generic technique however suffers from missing some important topics that are not mentioned rigorously in documents. In this regard, if topic modeling is included before preparing a summary, the probability of missing out on any topic will be minimized. Therefore, researchers today are mainly focusing their research directions to develop summarization systems to address all these challenges in producing the best summary possible.

In this study, we propose a generic, graph-based extractive multi-document summarization system and named it as Grapharizer. We have addressed some of the biggest problems that are faced by the summarization systems, which are, to maximize the coverage of all the topics, and to reduce redundancy. In our system, summaries are generated by plotting the sentences as word-graph that returns shortest sentences related to a topic in form of short paths in the graph. Graphs are prepared for each topic, therefore there is a representation of every topic in the summary, without compromising for the redundant appearances of the sentences of the same topic in the summary. The proposed system, depicted in Figure 4.1, is evaluated over the DUC 2004 dataset. This dataset contains a variety of topics, and for every topic, ten files are discussing different aspects of the same news event. The nature of the dataset fits exactly with the challenges that modern summarization systems face. We generated another dataset named recent news articles following same pattern to validate the results of Grapharizer. Our system gives comparable results to the baseline methods devised to solve the summarization problems.

Evaluation is performed using Recall Oriented Understudy for Gisting Evaluation, ROUGE henceforth, which is discussed in detail in section 4.3.2.

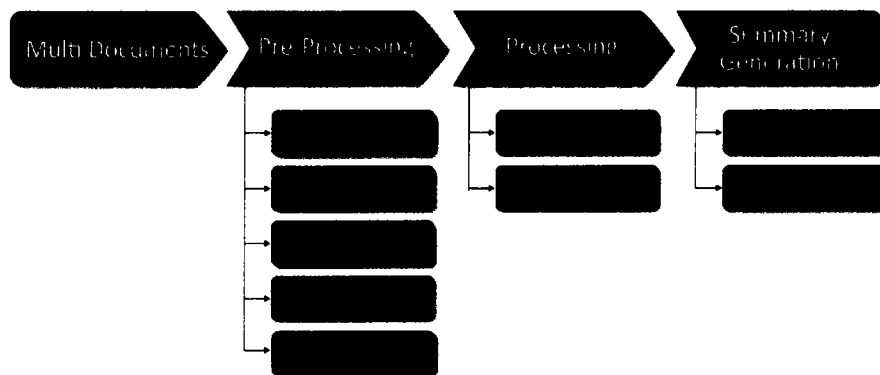


Figure 4.1: The Overview of the Grapharizer

The rest of the chapter is arranged as follows: in 4.1 we briefly discuss existing approaches related to text summarization, specifically the papers on the graph-based method of extractive summarization. Section 4.2 contains a detailed discussion on the proposed model as well as the steps taken to improve the grammaticality of the resultant summary. It also describes the machine learning methods adapted to improve the summarization system in the proposed method. Section 4.3 contains the details of the experiment conducted and the discussion on the results due to the variations in treatment to the model. In the end, we conclude our study in section 4.4.

4.1 Background

There are many techniques to facilitate the extractive MDS tasks, like Ontology-based [16, 38, 39], Term-based [4, 7, 15, 20, 43, 57], Clustering-based [21, 53, 63, 50, 60, 62, 66], Latent Semantic Analysis [35, 45, 47, 58, 59, 64, 67, 68], and Graph-based [46, 30, 29, 69, 48, 49, 51, 54, 56, 65] technique. Since our proposed technique, Grapharizer is producing summaries from input text using the graph technique, therefore we will present the related literature in detail in this section.

The use of graph theory for generating summaries is not new. It has its roots deep down in the field of ATS. Erkan and Radev in 2004 [30] used the graph theory first time for summarization. They devised LexRank algorithm which was the adaptation of PageRank algorithm. The sentence salience in a graph is computed using eigenvector centrality to compute the

importance of the node in the sentence-graph. The algorithm was tested over variants of DUC and evaluated via ROUGE.

Once the graph theory worked successfully in MDS, then a lot of work started in this field. Baralis et al. [48] adapted Apriori technique of Association Rule Mining in MDS to find the correlation between terms in the datasets and ranked the sentences then using PageRank [35]. Christensen et al. [65] used directed graphs for sentence selection and order to generate coherent summaries. Sukumar and Gayathri 2014 [56] worked to achieve sentence ordering by entailing method employing WordNet, alongside the graph theory. John and Wilsy [51] incorporated Euler's graph theory and named it Vertex Cover algorithm for MDS. Sentences were mapped into vertices. The vertices with high relevance scores qualify to appear in the summary.

ShafieBavani [46] compressed the sentences through word-graph technique yet kept up with the grammaticality of the summaries extracted from the unstructured dataset. They also mapped MWEs with their single word equivalents whereas Canhasi [69] presented a five-layered heterogeneous graph for MDS. Tzouridis et al. [54] presented word-graph method and compressed similar sentences by finding the shortest path in the graph. Machine Learning was also used to classify the candidate sentences in graph for the summary.

The idea of summarization has also attracted researchers from domains other than text. For instance, it has been adopted heavily in computer vision. Chen et al. [91] used the deep learning technique of CNN and RNN to summarize the multi-modal text. They have created an image summary from the source file and aligned the sentences and images in the extracted summary. They extended the DailyMail/CNN dataset by adding images to test the algorithm for multi-modal text corpus. The system was evaluated on ROUGE metrics and promising results were reported. Similarly, Celikkale et al. [92] adopted the idea of automatic summarization to generate structural summaries for a huge collection of images. The graph of images was constructed by predicting the next image approach. Image and textual data are used alongside the time and geographical information to manage the summary graph in order. Special dataset YFCC100M-CITIES was created to test the summarization system.

Shingrakhia et al. [93] tried to extend the benefits of summarization to the cricket grounds. They have devised a technique to generate a cricket match summary by finding exciting audios and then the corresponding visuals of fours, sixes, wickets, brilliant shots, bowls, fielding, umpire decisions, and challenges between players.

Like the entertainment industry, security is also one of the important domains in our lives. Summarization attracted the researchers to benefit the security domain too. Radarapu et al. [94] devised a technique to check the security camera videos in playback motion, or any movements in frames to any criminal activities in a short time and produce a summarized video clip for reporting purposes.

ATS is hugely used to check the opinion of people about certain subjects [89, 95, 96]. Nathania et al. [89] devised a technique to check for the summary of hotel reviews before booking by the tourists. Marzizarani and Sajedi [95] carried out the same task by using the clustering-based method.

When it comes to reviews, one just cannot ignore the significance of movie reviews. Abdi et al. [96] proposed a technique to summarize the movie reviews on the MovieReview dataset of IMDB using deep learning. The technique was also tested on DUC dataset variants to evaluate their technique generally too.

Bidoki et al. [85] devised a technique for extractive MDS employing statistical, graph, and machine learning techniques to optimize the resultant summary. They have focused specifically on the improvement of extraction of the short sentences in text, by expanding them with appropriate suitable words and tuning the overall sense in them.

Cross-language platforms are also adopted in ATS. For example, the documents' cluster is in an unfamiliar language to the user, so instead of translating the entire cluster, it is a wiser approach to tell the gist of it in the target language to the user. Pontes et al. [97] handled this issue with promising results. Compressive approach was applied at multiple as well as single sentence level so that a French-to-English summary could be generated. The one big objection to this study is for translations they have relied on Google Translate System, which usually fails to translate the MWEs across the languages.

El-Kassas et al. [98] proposed a mixed approach of a graph, statistical, semantic, and centrality-based methods, and named it EdgeSumm. They proposed a combination of four algorithms to resolve the task of summarization. They construct text-graph, then scan the graph for summary candidate selection, and then adjust it up to the set limits of length. The EdgeSumm claimed comparable results to state-of-the-art algorithms.

KUSH summarization system by Uçkan et al. [90] used the sentence relations to maximum levels. They exploited the idea of independent sets to analyze the graph nodes to select the most

representative sentence into summary, containing maximum information as well. Sentences are ranked and selected using Eigenvector Centrality. An increase in informativity and a decline in redundancy were reported in results.

Another attempt to discover cross-sentence relations was done by Wang et al. [99] in the technique named HeterSumGraph. They constructed sentence-graph and enriched it with semantic nodes at different levels. Sentences were connected with semantic nodes that were constituted using words in the sentences. The graph was scalable through the multi-documents by introducing document nodes. The technique can be extended for further better results by extending the features to topics and entities.

Tomer and Kumar [100] employed a bio-inspired firefly metaheuristic algorithm with improved fitness function to evaluate the features, to generate the summary with highly relevant sentences concerning the given topic. Based on the attraction of less bright fireflies towards the brighter ones, this algorithm was utilized in MDS as moving non-optimal solutions towards the optimal ones. The algorithm performed better ROUGE 1 and 2 scores on DUC dataset variants in comparison to other nature-inspired algorithms.

Davoodijam et al. [101] proposed an unsupervised multi-layered graph-based system to summarize the medical documents and named their system as MultiGBS. The multiple layers of an undirected graph were found useful to cover multiple features. Employing the features, sentences were ranked using MultiRank algorithm, a PageRank-like algorithm to rank sentences of a multi-layered graph. Evaluation of ROUGE and BERTScore give better F-measure scores. The MultiGBS needs to be evaluated on benchmarked datasets to best understand the claimed results on medical documents.

Jin et al. [102] adopted a hierarchical relation graph-based interaction network technique that treats documents, phrases, and words at various levels for extractive and abstractive summarization. The system was evaluated using the Multi-News dataset, and the results are encouraging.

Lierde and Chow [103] presented a query-focused ATS system based on hypergraph, where sentences were mentioned as graph-nodes and hyperedges combine the nodes of similar topics. Whereas Li et al. [104] argued that the summarization systems either fail to contain the sentiments in the summary, or it gives wrong sentiments from the original document sets altogether. They have used the graph theory for summarization as well as maintaining the

sentiment vector for the sentences to keep the sentiments of the documents in the summary, intact.

The major motivation for developing Grapharizer is to reduce the negative consequences of existing ATS systems. To leverage from their advantages and overcome the drawbacks of each method individually, Grapharizer combines its grammar-focused pre-processing with LDA algorithm of topic modeling, and used graph-based methodology with SVM technique of machine learning as follows:

- Proposing a supervised and domain-independent framework that does not require data to be present in annotated form for processing and is not limited to documents from a single domain.
- Using the LDA topic modeling technique to create a summary that includes all the subjects in the input document.
- The proposed methodology increases the informativity, representativeness of the summary by including all the topics in the resultant summary and removing the redundancy in it using topic modeling and generating graphs for every topic individually and combining the summaries of all topics in the end.
- Use of Lemmatization, MWE mapping, synonym mapping with novel cross function, pronoun replacement in anaphora and cataphora in pre-processing, and reverse MWE function in summary aim to increase the grammatical quality of the resultant summary

4.2 The proposed technique

This section contains the introduction of the graph-based technique that we have proposed for EMDS named Grapharizer. After the motivation of our proposed technique, we elaborate the graph creation process of our technique, how it is designed to handle the problem of redundancy, and how it is made not to miss any important topic in the given set of documents while preparing the summary.

4.2.1 Motivation

Graph theory is an important structure to solve the optimization problems in every important field of Science, Technology, Engineering, and Mathematics, or STEM in short. It has been extensively used to solve different problems like finding shortest path, connectivity among the different networks like computer networks, social networks, academic link networks, and

connectivity networks, and detecting malicious activities from inside the networks to avoid the cyber-crimes.

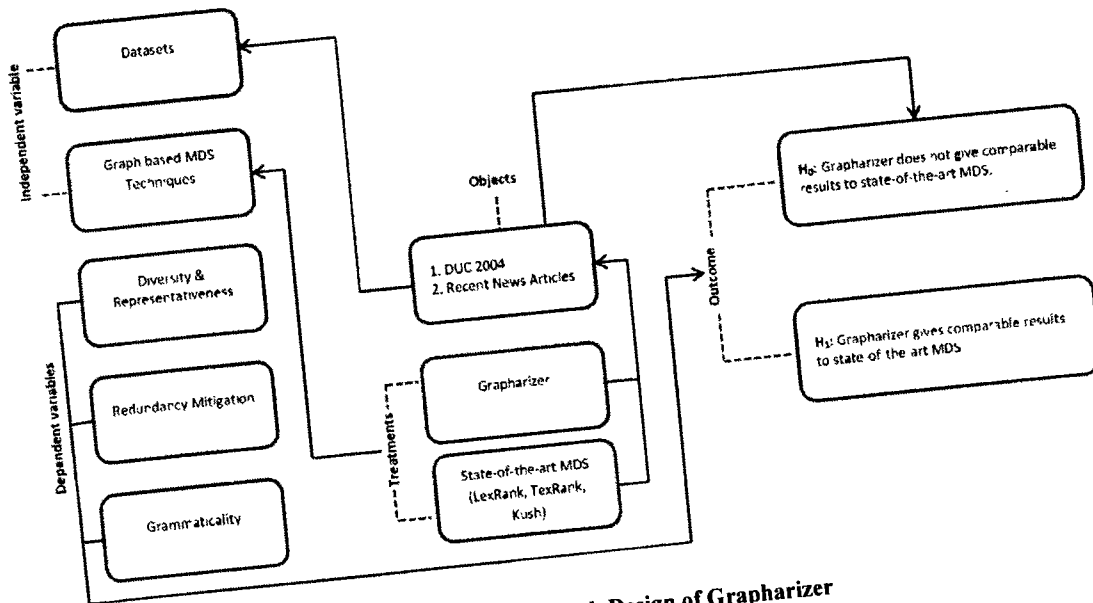


Figure 4.2: The Research Design of Grapharizer

Graph theory has been extensively utilized in the field of ATS for the last 25 years. It has, however, few limitations. For example, the edges between two busy vertices can be considered similar [105] which might be quite different in nature. It can possibly miss the important content from the documents while stressing the same point redundantly in the summary.

In this experiment, there are two independent variables: Graph-based MDS Techniques and Datasets. For each independent variable, there are two treatments. The Graph-based techniques consist of two types: Grapharizer and State-of-the-art MDS (LexRank, TexRank, Kush). The datasets used in the experiment are DUC 2004 and Recent news articles. The dependent variables being measured are diversity and representativeness, redundancy mitigation, and grammaticality, as shown in Figure 4.2. To ensure consistency, the effect of the dataset is blocked, and the same datasets are used for both treatments of the Graph-based MDS techniques in each trial of the experiment.

4.2.2 Grapharizer: The Graph-Based Method

As discussed in section 4.2.1, the graph theory is integral in ATS to produce the optimized summaries. The critical problem in extractive multi-document summarization is to find out the

important sentences from the input text. It is also important to note that the multiple documents might be containing different topics, some of the topics are frequently discussed while some topics are discussed less frequently. It is nevertheless important to cover all the aspects present in the documents to prepare a generic summary, to get the gist of the documents appropriately. The graph-based methods used in MDS generate the graph based on the input document (we will discuss it in subsection 4.2.2.2) and find the shortest path in the graph to generate the summary. That short path might cover only some of the topics, not all the important ones, hence resulting in a less representative summary. We have handled this problem by incorporating the topic modeling before generating the graph.

Similarly, in Extractive MDS, the grammaticality of summary always suffers. The main areas of concern are pronouns, or the anaphora and cataphora. The sentences containing noun mentions might not be selected in summary, but pronoun mentions might get a place there. This can give serious confusion to the reader of the summary. Similarly, since the input data is always unstructured text, it might contain the phrases like idioms to explain a situation poetically. Such phrases, or MWEs, can cause concerns about the length of the summary. Also, in the word graph-based methods (explained in section 4.2.2.2), there is a possibility of picking a few vertices from the MWEs and dropping the rest, which results in a meaningless summary. Therefore, we have handled this problem by replacing MWEs with their single-word synonyms.

4.2.2.1 Pre-processing

Since the NLP projects are dealing with unstructured data, therefore it is unavoidable to skip the step of pre-processing. It prepares the data for the algorithm to facilitate summary generation. The better the pre-processing is, the more accurate will be the results generated by the algorithm.

We have applied several steps in pre-processing phase to convert the input data to work well with the Grapharizer. The steps performed in pre-processing to cleanse the input text for our summarizer are stated below:

Tokenization: in tokenization, each sentence from the input documents is broken down into the smallest units, called the Tokens. This is helpful to recognize the words for the later functionalities in the subsequent pre-processing steps and graph generation module.

connectivity networks, and detecting malicious activities from inside the networks to avoid the cyber-crimes.

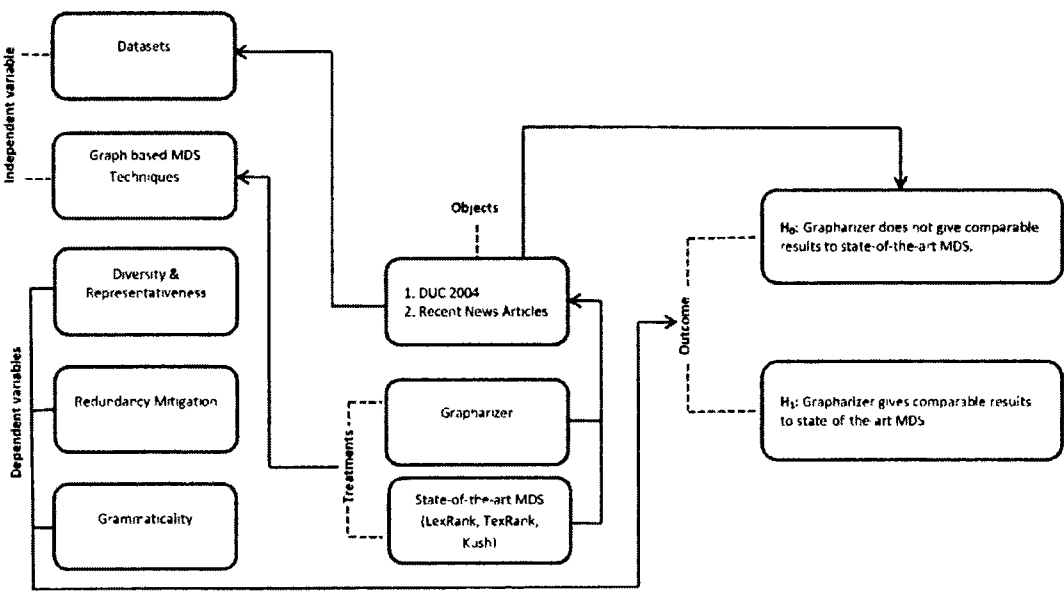


Figure 4.2: The Research Design of Grapharizer

Graph theory has been extensively utilized in the field of ATS for the last 25 years. It has, however, few limitations. For example, the edges between two busy vertices can be considered similar [105] which might be quite different in nature. It can possibly miss the important content from the documents while stressing the same point redundantly in the summary.

In this experiment, there are two independent variables: Graph-based MDS Techniques and Datasets. For each independent variable, there are two treatments. The Graph-based techniques consist of two types: Grapharizer and State-of-the-art MDS (LexRank, TexRank, Kush). The datasets used in the experiment are DUC 2004 and Recent news articles. The dependent variables being measured are diversity and representativeness, redundancy mitigation, and grammaticality, as shown in Figure 4.2. To ensure consistency, the effect of the dataset is blocked, and the same datasets are used for both treatments of the Graph-based MDS techniques in each trial of the experiment.

4.2.2 Grapharizer: The Graph-Based Method

As discussed in section 4.2.1, the graph theory is integral in ATS to produce the optimized summaries. The critical problem in extractive multi-document summarization is to find out the

important sentences from the input text. It is also important to note that the multiple documents might be containing different topics, some of the topics are frequently discussed while some topics are discussed less frequently. It is nevertheless important to cover all the aspects present in the documents to prepare a generic summary, to get the gist of the documents appropriately. The graph-based methods used in MDS generate the graph based on the input document (we will discuss it in subsection 4.2.2.2) and find the shortest path in the graph to generate the summary. That short path might cover only some of the topics, not all the important ones, hence resulting in a less representative summary. We have handled this problem by incorporating the topic modeling before generating the graph.

Similarly, in Extractive MDS, the grammaticality of summary always suffers. The main areas of concern are pronouns, or the anaphora and cataphora. The sentences containing noun mentions might not be selected in summary, but pronoun mentions might get a place there. This can give serious confusion to the reader of the summary. Similarly, since the input data is always unstructured text, it might contain the phrases like idioms to explain a situation poetically. Such phrases, or MWEs, can cause concerns about the length of the summary. Also, in the word graph-based methods (explained in section 4.2.2.2), there is a possibility of picking a few vertices from the MWEs and dropping the rest, which results in a meaningless summary. Therefore, we have handled this problem by replacing MWEs with their single-word synonyms.

4.2.2.1 Pre-processing

Since the NLP projects are dealing with unstructured data, therefore it is unavoidable to skip the step of pre-processing. It prepares the data for the algorithm to facilitate summary generation. The better the pre-processing is, the more accurate will be the results generated by the algorithm.

We have applied several steps in pre-processing phase to convert the input data to work well with the Grapharizer. The steps performed in pre-processing to cleanse the input text for our summarizer are stated below:

Tokenization: in tokenization, each sentence from the input documents is broken down into the smallest units, called the Tokens. This is helpful to recognize the words for the later functionalities in the subsequent pre-processing steps and graph generation module.

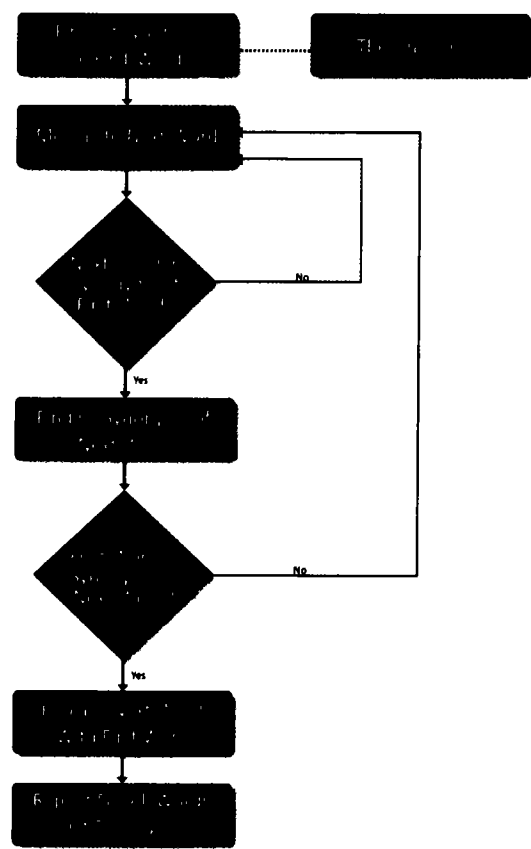


Figure 4.3: Working of the Cross() for synonym mapping

Lemmatization: in this step, the given words/tokens are reduced to their root words, called lemmas. It sounds similar to stemming which simply chop off the words from the tail, while in lemmatization the words are reduced to the root word with proper meaning using some dictionary. The reduced words in lemmatization are meaningful grammatically. Since grammatically correct summaries are one of our aims, therefore lemmatization is close to our goals in summarizer.

Synonym Mapping: Since the Grapharizer is based on the word graph, the graph it constructs based on multiple documents is going to be too dense. It is important to replace the synonyms to keep the graph simple. Therefore, synonym mapping is the most important step in pre-processing of our system to make the constructed graph under control. We have used Thesaurus.com for synonym mapping. There are several synonyms thesaurus.com offers against a particular word. We have devised a function named cross() to pick the best matching synonyms. For example, we have a word A in the document, and then comes another word B.

As shown in Figure 4.3, we must check whether B or its synonyms already exist in the document or not. We will check the synonyms of A. If B exists in the synonym list of A given by thesaurus.com, we will check conversely too by checking the synonyms of B. Thesaurus.com will give a list of synonyms of B as well. Now, if A is also the synonym of B, then the condition of `cross()` is satisfied, and B can be replaced with A.

Multi-Word Expressions: As discussed in section 4.2.2, the MWEs are likely to appear since the input documents are unstructured files, therefore we have given special care to map the MWEs with their single word synonyms. This will not only bring simplicity to the graph, and it would appear clearer and more manageable. Once the candidate sentences for summary are selected, we apply a function `reverse-mwe()` that we have devised to reverse the single worded synonyms replaced earlier back to their corresponding MWEs. This will result in more close results to the gold standard summaries of the benchmark datasets, like DUC 2004 in our case.

Pronoun Replacement: in this step, the pronouns are replaced with their corresponding nouns. As stated earlier, in a document, some sentences are mentioning the nouns about certain events, and later, they can be referred to by pronouns. This situation is also known as Anaphoricity. Sometimes, it also happens that the pronouns are mentioned earlier, and the nouns follow them later in the text. This situation is known as Cataphoricity. There is a possibility that the sentences with noun mentions are not extracted as important fragments of the text, while the sentences with pronouns are selected. This can cause serious confusion in the summary that what these pronouns are referring to. To solve this problem, we have used the pronoun replacement. This phase is mainly inspired by the Durrett et al 2016 [42].

4.2.2.2 Overview of The Graph Generation Process

We have used the word graph to generate the short paths in the multiple documents [54, 82]. The process to construct a graph begins after the pre-processing phase. The first vertex of the graph is vertex S, the start vertex of the graph. It then starts reading the sentences in sequence. The first word of the first sentence of the first document will be placed as a graph vertex. There will be a directed edge from the S vertex to the vertex of the first word of each sentence. Then for the second word, an edge will be directed from the first word towards the second word, and so on. In this way, all the sentences are plotted in the graph, word by word. Each word makes a vertex in the graph. If a sentence contains a word that is already used in some previous sentences, then that word will already be placed as a vertex in the graph. Therefore, there is no need to duplicate that vertex. An edge will be directed towards that vertex in the graph. This

process continues until the last word of the last sentence in the last document is plotted as a graph vertex. It is important to note that every sentence in the document end at the E, the end vertex of the graph. The in and out degree of each vertex is calculated. This working is explained with an example in Figure 4.4.

Once the graph is constructed, now the aim is to find the shortest path in that graph. To find the shortest path, we used Dijkstra's algorithm. It finds the shortest path between vertices S and E. The lesser the vertices traversed, the shortest the path is.

To find k-short paths in the graph, we have employed Yen's algorithm as used by Tzouridis et al [54]. We have optimized the summaries by incorporating a few constraints. For example, if there are no vertices between S and E vertices, such short paths, also known as empty paths, will be discarded to qualify as summaries. Similarly, if a sentence does not contain noun, predicate, and verb, then it is not a candidate sentence for a summary. We have also set a certain limit to the size of the summary like it should not be less than 15% of the input text. Too much compression can also result in loss of meaning to the summary.

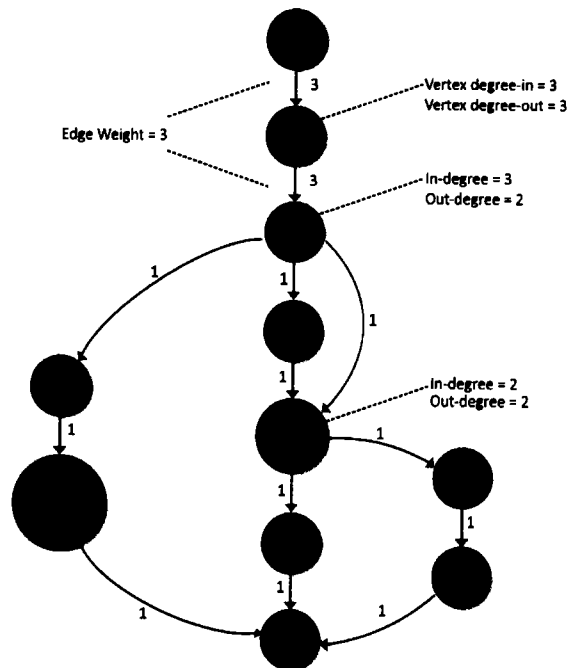


Figure 4.4: Word graph construction from the given sentences.

"Linda is a beautiful girl", "Linda is a beautiful young lady", "Linda is very professional". The shortest path is the summary of these sentences in red highlighted path saying, "Linda is beautiful girl".

4.2.2.3 Representativeness

It has been observed that the datasets contain plenty of unstructured information about different topics. The frequency of some topics appears to be higher than the others. A fully representative summary should have complete coverage of all the topics in the documents and should not miss any important topic to be mentioned in the summary, as this is the basic idea of generic summarization. As mentioned earlier, it is important not to miss any topic from appearing in the summary. Therefore, we have handled this issue by employing topic modeling. We have applied Latent Dirichlet Allocation LDA [106], Hierarchical Dirichlet Process HDP[107], and Latent Semantic Indexing LSI [108] to see the effect of topic modeling over the results of the summarization process. LDA improved the results more, so we have used it in our model.

Topic modeling with graph works in this way: different topics in the input document are identified with the help of LDA algorithm. Then the sentences of the same topic are combined across the multiple files in the dataset. Then, for each topic, the graph is plotted, and the shortest paths are identified based on the mechanism discussed in section 4.2.2.2. Once the short paths are identified in all the graphs, or topics, then they are combined based on the criteria discussed earlier. The resultant summary does not miss any topic from the given dataset, hence a truly presentative generic summary.

4.2.2.4 Removing The Redundancy

The repeated appearance of the same topic again and again in the summary results in poor quality of the summary. A user receives less information from the summary due to this redundancy problem. This is one of the major concerns of extractive MDS [81].

As shown in Figure 4.5, we have addressed this problem in our Grapharizer. We employ topic modeling to identify different topics and construct their graphs separately. The selected shortest paths from each topic will have their appearance in the summary in the end. Therefore, the resultant summary will not be bombarded by the same set of sentences, rather it will have representation from each topic corresponding to its volume in the dataset. This way, the redundancy problem has been handled in Grapharizer.

4.2.2.5 Grammaticality

The abstractive summaries are having an edge over the extractive summaries for their grammar quality. Since the abstractive summaries are written after the documents are read in words not like that in the datasets, they do not suffer from grammar quality. The short new sentences are produced keeping in mind the grammar rules in language generation. Extractive summarization, however, does not function in this manner. Since it picks the extract from the input document cluster, therefore there can be a huge compromise on the grammar quality once the summary is generated. As discussed in section 4.2.2.1, in the pronoun replacement subsection, if the sentences containing noun phrases are skipped to be candidates in the summary, and sentences containing pronouns are somehow selected in the summary, it will

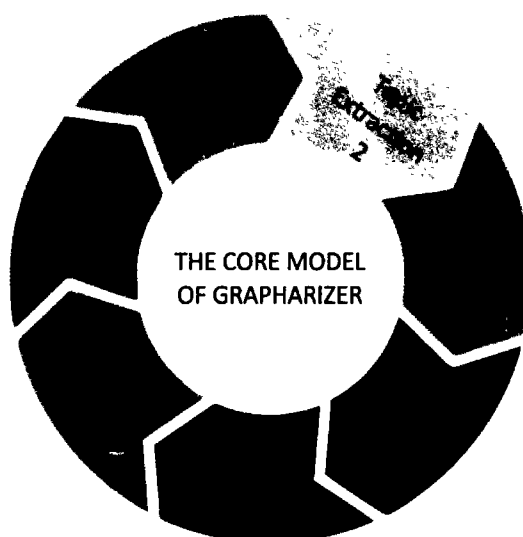


Figure 4.5: The core model of Grapharizer

possibly be a confusing summary for the reader. Instead of facilitating the user, such summarization systems lead to confusing the users. Therefore, we have taken special care of the grammar quality of the summary. The candidate sentences will be corrected grammatically before appearing in the summary. Synonyms and MWEs are mapped, anaphora and cataphora constraints are resolved before the application of Grapharizer, resulting in grammatically correct summaries. To retain the poetic touch in the summary, all the MWEs that were replaced previously will be re-applied before presenting the summary to the user. Similarly, we have preferred lemmatization over stemming to trim the words with proper meaning, not just chopping off from the end.

4.3 Experiment and Evaluation

The details of the experiment conducted, and its evaluation is presented in the following sub-sections.

4.3.1 Dataset

National Institute of Standards and Technology (NIST) runs a series of conferences every year to evaluate the summarization systems' performance. These conferences are conducted under the title of Document Understanding Conference (DUC). DUC is the platform that encourages and facilitates researchers in their scientific inventions and progress.

DUC occasionally releases standardized datasets for document summarization, so that the researchers can be provided with similar input document(s) to compare the progress of their summarization system with the others. These datasets contain reference summaries as well that are made by human assessors. Reference summaries are also termed as gold standard summaries by some researchers. ROUGE evaluation is applied to evaluate the system-generated summary against the reference summary. DUC 2001, DUC

2002, DUC 2004, DUC 2005, and DUC 2007 are the different versions of the datasets released in the years mentioned in the name. Some datasets contain the reference summaries that are made generically, while some are query-focused.

DUC 2004 contains 50 folders; each folder contains 10 documents about different aspects of the same related news item. Each folder out of 50 is equipped with 4 reference summaries so that when a summarization system produces a summary, it could be evaluated against it to gauge the accuracy and precision of the system.

Similarly, we have created another dataset with similar features to DUC 2004 and named it Recent News Articles dataset. This dataset contains ten files regarding same news event (flood in Pakistan 2022, Divorce news of Johny Depp and Amber Heard, Death of Queen Elizabeth II, etc.). Like DUC 2004, we have provided four reference summaries to compare the system summaries for performance evaluation. In this regard, we have used ChatGPT to prepare reference summaries. The summary length was kept under the word limit of 200 words in order to keep the consistency between system and reference summaries.

It is worth mentioning that except for DUC 2005 and DUC 2007, the other variants are generic in nature concerning the reference summaries provided.

We have tested our Grapharizer on the DUC 2004 and Recent News Articles datasets, and for evaluation, we have used the ROUGE score. The major characteristics of the DUC 2004 and Recent News Articles datasets are shown in Table 4.1.

Table 4.1: Characteristics DUC 2004 and Recent News Articles at a glance

Characteristics	DUC 2004	Recent News Articles
Total topics	50	25
Number of documents (per topic)	10	10
Total number of documents	500	250
News/ data source	TRD, TREC collection newswire	News articles

Apart from DUC, there are other benchmarked datasets as well. New York Times annotated dataset, TAC, SemEval, MiltiLing are a few examples.

4.3.2 Evaluation Metric

DUC has included the Recall Oriented Understudy for Gisting Evaluation, or ROUGE, as its most valued evaluation metric to gauge the quality of the system generated summary against the reference summaries as well as the machine translations. It counts the number of similar units between the system summary and the reference summary. It employs Recall, Precision and F-Measure in context of ROUGE. Considering N as length of N-Gram, Countmatch (N-Gram) refers to overlapping words between the system summary and the reference summary, and Count(N-Gram) representing the words in reference summary, the ROUGE-N can be computed by the following formula:

$$\text{ROUGE-N} = \frac{\sum_{S \in \text{Reference-Summary}} \sum_{N\text{-gram} \in S} \text{Count}_{\text{match}}(N\text{-Gram})}{\sum_{S \in \text{Reference-Summary}} \sum_{N\text{-gram} \in S} \text{Count}(N\text{-Gram})} \quad (5)$$

In our experiments, we have used all the metrics of ROUGE, but for the simplicity of the results, we are mentioning just ROUGE 1, ROUGE 2, ROUGE L, ROUGE W, and ROUGE SU*.

4.3.3 Evaluation and Comparison

We have evaluated Grapharizer with state-of-the-art graph-based methods for comparison. The methods chosen are discussed:

TextRank: Mihalcea and Tarau [112] created TextRank, an iterative, extractive, and unsupervised tool that grades units based on the relevance of text units, by translating the connected structure of a text to graphs. The TextRank method, which is iterative research based on Google's PageRank, has no predefined and manually configured structure, which is a key characteristic.

LexRank: To calculate the importance of textual units in NLP, Erkan and Radev [30] introduced a stochastic and graph-based technique called LexRank. They used the eigenvector centrality (node centrality-based) measure to calculate the importance of sentences on the representative graph. In their experiments, the authors found that LexRank outperformed both degree-based and centroid-based algorithms in many cases.

Kush: According to the assumption by Uçkan and Karcı [90], the summary should exclude the set of sentences matching the nodes in the independent set. The nodes composing the Independent Set on the graphs were identified and eliminated based on this prediction. Resultantly, the restriction was placed on the documents to be summarized before quantifying the effect of the nodes on the global graph. That limitation made it impossible to include word groups in the summary that were repeated.

The experimentation of Grapharizer was implemented using intel core i7 system and the implementation was done using Python. Many functions were created to attain the pre-processing, and the processing subsequently. The cross function was devised specifically to return the most optimal synonyms for terms. Similarly, the library for MWEs was designed for the both the datasets to generate a convenient word graph. The MWEs replaced in the graph generation process to its single word equivalents may result in a poor ROUGE score when evaluated against the reference summaries, therefore we have developed another function named `reverse-mwe()` to replace the single worded equivalents of MWEs with the original MWEs.

We perform ablation study to verify the effect of the different parts of pre-processing over the overall performance. Five stages of ablation were compared with our results. First, we removed the pre-processing completely. The results significantly declined as mentioned in table 4.2. Then we added pronoun replacement module to our Grapharizer, which enhanced the results by a margin of 15%, as shown in table 4.2. After that, in the third stage, we included only the synonym mapping module. 20% improvement was reported in Rouge 1, consequently. In the fourth stage, we applied MWE (without `reverse-mwe()` of post-processing) and nearly 15%

gain was reported in results of Rouge 1. Finally, we tested MWE with reverse-mwe() in post-processing phase. The improvement was 40% in Rouge score. When entire pre-processing was applied and results were compared stage 1 of ablation, i.e., no pre-processing, then reported results display 70% improvement in Rouge scores. Therefore, it is safely concluded that the pre-processing and post-processing contributed positively to our technique, Grapharizer.

Table 4.2: Ablation study representing the effects of different pre-processing phases on Grapharizer.

	Without Pre-processing			
ROUGE-Type	Recall	Precision	F-Score	Improvement
Rouge-1	0.12567	0.16509	0.14271	-
Rouge-2	0.00368	0.005	0.00424	-
Rouge-L	0.13274	0.16111	0.14555	-
Rouge-SU4	0.03745	0.04251	0.03982	-
	With pronoun replacement			
Rouge-1	0.18188	0.19841	0.18979	0.04708
Rouge-2	0.00831	0.00806	0.00818	0.00394
Rouge-L	0.15956	0.17857	0.16853	0.02298
Rouge-SU4	0.05296	0.04836	0.05056	0.01074
	With synonym mapping			
Rouge-1	0.15872	0.1822	0.16965	0.02694
Rouge-2	0.02082	0.02155	0.02118	0.01694
Rouge-L	0.15627	0.15741	0.15684	0.01129
Rouge-SU4	0.04616	0.04474	0.04544	0.00562
	With MWE mapping			
Rouge-1	0.17216	0.19915	0.18467	0.04196
Rouge-2	0.01703	0.01724	0.01714	0.0129
Rouge-L	0.1612	0.16827	0.16466	0.01911
Rouge-SU4	0.05045	0.04912	0.04978	0.00996
	With reverse-mwe()			
Rouge-1	0.32949	0.24722	0.28249	0.13978
Rouge-2	0.09168	0.0618	0.07383	0.06959
Rouge-SU4	0.11458	0.07159	0.08812	0.0483

After the pre-processing stage, we have applied the LDA algorithm of topic modeling to separate the sentences related to a different topic and combine them under respective topics. Then, we have applied the Grapharizer over the individual topic. Related sentences were plotted as a graph using the word graph technique adapted from Filippova [82] and Tzouridis et al. [54]. Once the graphs of each topic were generated, the shortest paths in the graphs were calculated using the Dijkstra algorithm. Care has been taken in selecting the shortest paths as these paths were the sentences of prospected summary. It has been decided to keep the paths from every graph as summary sentences that included noun, predicate, and verb.

Compared with different state-of-the-art summarizers, there was a significant improvement in the Rouge scores, as shown in Table 4.3.

Compared with TextRank [112], Grapharizer gained 6.24% accuracy in Rouge 1 recall. Compared with LexRank [30], Grapharizer improved the Recall assessment of Rouge 1 by almost 14%. Compared with Kush [90], Grapharizer improved the Recall of Rouge 1 by 8.06%.

Table 4.3: Comparison of Grapharizer With State-of-The-Art Graph-Based Methods

ROUGE evaluation		Methods				Grapharizer on Recent News Articles
METRICES		TextRank	LexRank	KUSH	Grapharizer on DUC 2004	
<i>Rouge 1</i>	Recall	0.39893	0.32206	0.38072	0.46136	0.42018
	Precision	0.33462	0.30458	0.34019	0.21030	0.13863
	F-Score	0.36292	0.31255	0.35879	0.28891	0.20848
<i>Rouge 2</i>	Recall	0.07977	0.05439	0.08277	0.09822	0.04846
	Precision	0.06831	0.05170	0.07373	0.04203	0.01484
	F-Score	0.07338	0.05292	0.07786	0.05886	0.02273
<i>Rouge L</i>	Recall	0.30685	0.25785	0.29037	0.17050	0.14574
	Precision	0.25868	0.24444	0.25945	0.08979	0.06627
	F-Score	0.27991	0.25053	0.27364	0.11763	0.09111
<i>Rouge W</i>	Recall	0.10436	0.08957	0.09908	0.10299	0.10299
	Precision	0.16113	0.15517	0.16197	0.08422	0.08422
	F-Score	0.12622	0.11334	0.12278	0.09238	0.09238
<i>Rouge SU</i>	Recall	0.13998	0.09532	0.13062	0.18137	0.19608
	Precision	0.09683	0.08421	0.10372	0.06751	0.06309
	F-Score	0.11328	0.08884	0.11493	0.09840	0.09547

The results present very interesting picture of the performance of different graph-based methods. As shown in the figure 4.6-4.10, the algorithm performed well in Rouge SU for Recall, while in Rouge 1, it stood first in performance. Similar performance is evident in Rouge 2, where Grapharizer is leading the performance. In Rouge W, the Grapharizer is second best in performance while TextRank topped both Rouge W and Rouge L.

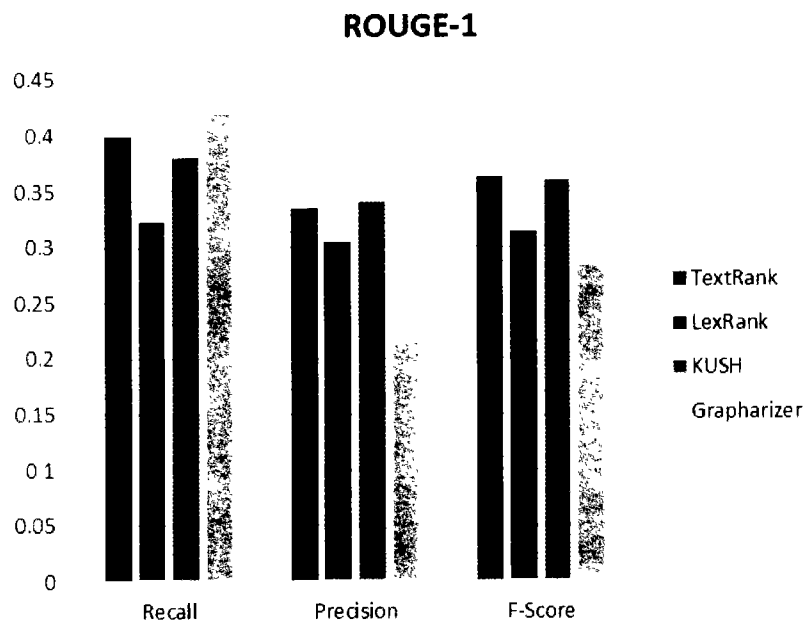


Figure 4.6: Grapharizer vs SOTA techniques for ROUGE 1

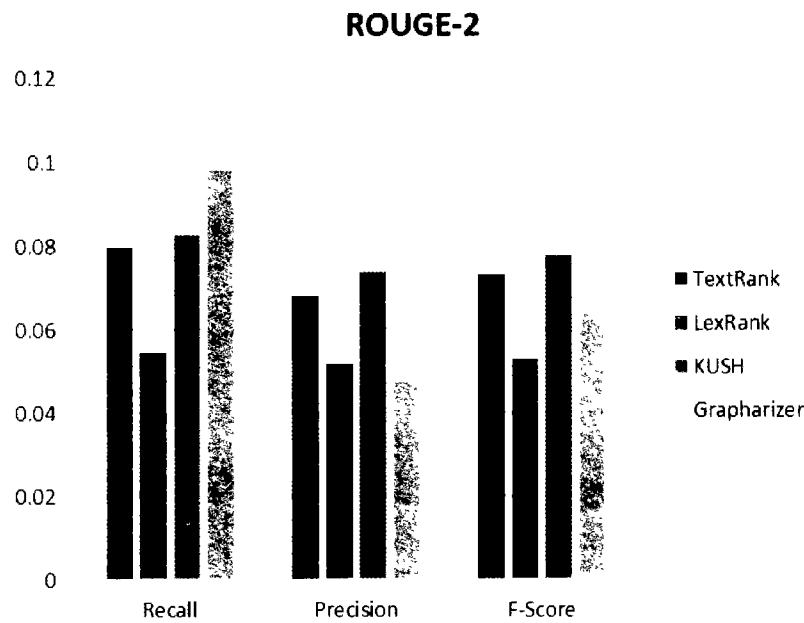


Figure 4.7: Grapharizer vs SOTA techniques for ROUGE 2

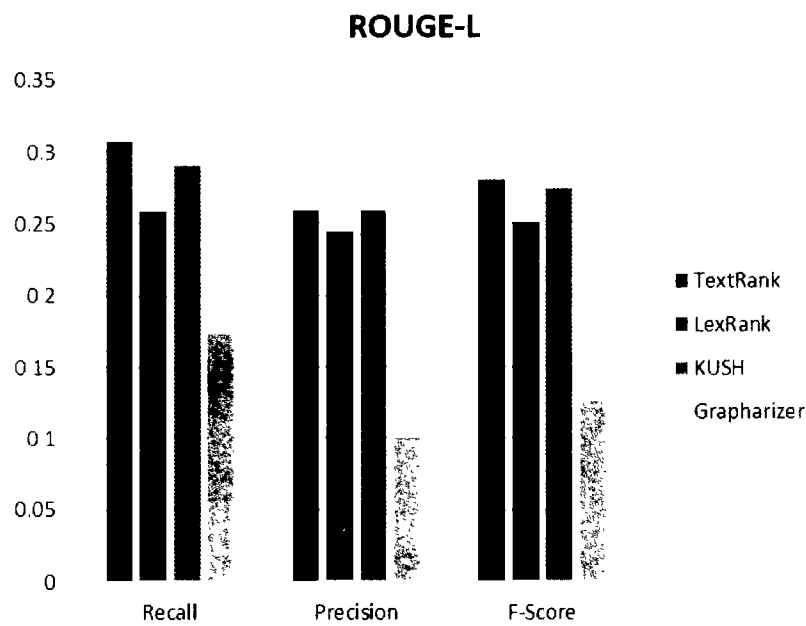


Figure 4.8: Grapharizer vs SOTA techniques for ROUGE L

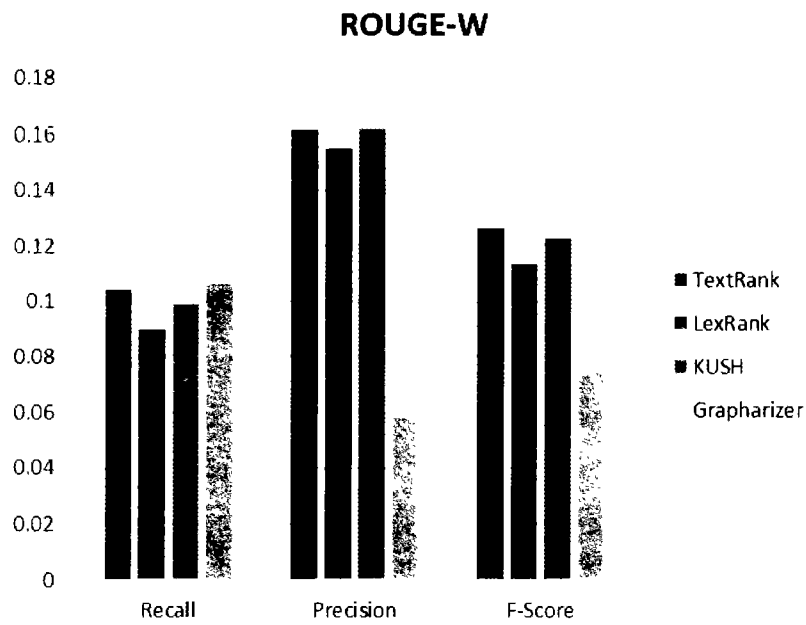


Figure 4.9: Grapharizer vs SOTA techniques for ROUGE W

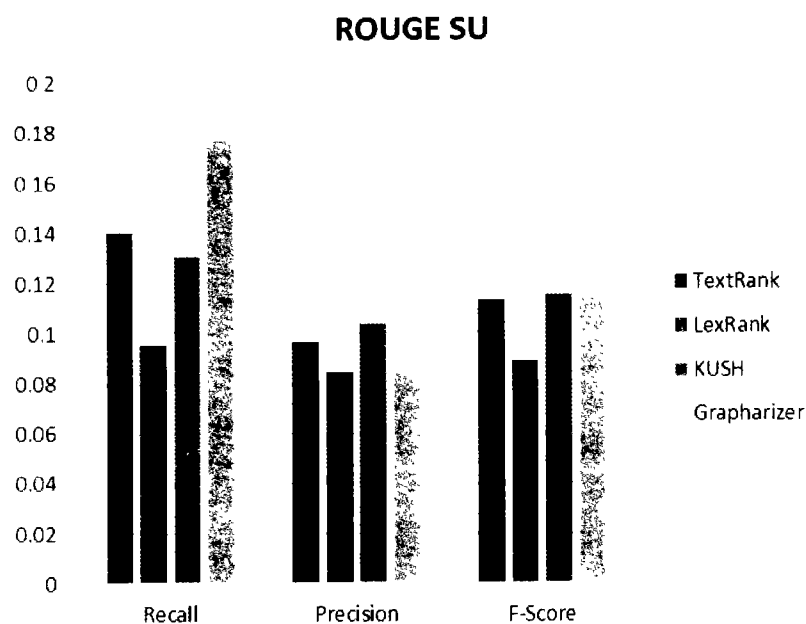


Figure 4.10: Grapharizer vs SOTA techniques for ROUGE SU

4.4 Conclusion

In the era of information overload, it is necessary to have access to accurate information in less time. The need is to have such concise accurate information provided against user queries that will not have redundant information, and it must contain all the aspects of the required subject.

In this chapter, we anticipated that the more data is prepared for processing in the pre-processing stage, the better would be the results. Furthermore, to avoid redundancy and to ensure the maximum coverage of all the topics of the input documents in the summary, we introduced the intermediate step of topic modeling before the actual processing. That resulted in a concise and informative summary that performed better than the different SOTA techniques.

To maintain the poetic flow of the documents reflected in the summary, we have applied the reverse-mwe() that not only served the said purpose but also helped in gaining improved Rouge scores in many variants of Rouge evaluation.

In subsequent chapters, we present the Grapharizer and its machine learning variants. Also, we present different evaluation techniques, like BERTScores and manual evaluation by using crowdsourcing platforms like Amazon’s Mechanical Turk, or UpWork. Grapharizer is also

being tested with different other summarization techniques that are not employing graph theory in order to check the best overall summarization technique.

We will improve the accuracy of `mwe()` and `reverse-mwe()` since this is one of the main reason of low precision score that also affected the f-score of various Rouge variants for Grapharizer.

Chapter 5

RST Model

5 Introduction

Rhetorical Structure Theory (RST) was initially introduced as a method for studying natural language text in discourse structure. It took a while for RST to be used in the automatic generation of text summary. Daniel Marcu [118] was the first to complete a viable RST-based discourse parsing and text summarising project. The concept of coherence is central to RST therefore it may also be thought of as a theory of text coherence.

RST is a theoretical framework for analysing text structure at the clause level. The method retrieves the text's rhetorical structure as well as the compound of rhetorical relations between phrases, then leaves out the less relevant parts. Finally, it employs a model-based natural language generation method to generate an accurate and fluent summary [119].

The rhetorical relations form the foundation of RST. Rhetorical relation exists between two sibling nodes that share a common parent node, with one child node acting as a nucleus (N) and the other as a satellite (S). Both children could possibly be nuclei. What is expressed in N is more important than what is conveyed in S in terms of the writer's goal. N is independent of S in terms of comprehension, but not the other way around.

The notion of rhetorical relation, which exists between two distinct text spans named nucleus (N) and satellite (S), is central to Rhetorical Structure Theory. Based on the empirical finding that the nucleus expresses what is more significant to the writer's goal than the satellite, there is a distinction between nuclei and satellites, therefore the nucleus of a rhetorical relation can be understood without the satellite. RST analysis entails creating a binary, tree-like structure to describe the text's coherence relations. This requires finding and classifying discourse unit relationships. These can be basic units that show up as leaves in the resultant tree, or they can be text spans that include two or more elementary units or lower-level spans [119].

According to Ibrahim et al. [120] the RST model can be divided into four steps.

- 1) RST Parsing Process: This method is used to determine the Elementary Discourse Unit (EDUs).
- 2) Rhetorical Relationship Identification Process: this step identifies the different rhetorical relations among the different EDUs.
- 3) Building the RS-Tree Process: After identifying the units and recognising the rhetorical relationships between them, the model then represents and construct a rhetorical structure using a binary tree topology known as the rhetorical structure tree (RS-Tree)

- 4) RST Selecting Process: The model then selects the RST by identifying the Nucleus (N) and Satellite (S) EDUs.

RST models help in extracting the semantics of sentences thus the context of the text is also conveyed through the summary.

5.1 The proposed technique

We have adapted the model by Durrett et al. [42] and extended it to multi-documents summarization. Previously, the model was tested on NYT annotated dataset. We have extended it to the unstructured dataset of DUC 2004.

We have applied the pre-processing in order to maximize the flow and cohesion in the summary and minimize the duplicated sentences from appearing in the summary. In order to do that, we have applied topic modelling on the unstructured text. Duplication of data is one of the major concerns in almost all fields of multi-document summarization. From each topic, the sentences are scanned for MWEs in order to simplify the text. As mentioned in chapter 3, the model begins its working by loading the multi-document dataset of DUC 2004. From all the documents, sentences are read, and discourse units are identified. Then the syntactic and rhetorical relations are applied. Once the relations are applied on the EDUs, the sentences are represented in binary tree model. The EDUs are then assigned as nucleus or satellite.

Since it is represented in binary tree format, therefore it contains directed edges from one tree node to other. What increases the cohesion in our RST based model is that we have taken good care of anaphora and cataphora in order to resolve the coreferences. We have applied the pronoun replacement at the pre-processing stage to avoid the appearance of conflicts at later stages.

As mentioned in chapter 4, we have applied the reverse-mwe() at the summary level, over the sentences selected for the summary. The results of our model are presented in table 5.1.

Table 5.1: ROUGE Scores of RST Summarizer

Rouge Metrics	Recall	Precision	F-Score
Rouge 1	0.53374	0.24567	0.33647
Rouge 2	0.15025	0.06522	0.09095
Rouge L	0.15732	0.08217	0.10795
Rouge W 1.2	0.11372	0.09742	0.10494
Rouge SU4	0.22342	0.09127	0.12959
Rouge SU*	0.26666	0.12669	0.17178

In previous studies, RST is applied for text summarization, no matter the single document summarization or multi-document, on annotated datasets. To the best of our knowledge, no study is conducted using RST on the unstructured dataset of DUC 2004. We have used an automated tool named RSTTool version3.3 (www.wagsoft.com/RSTTool/) to apply rhetorical relations among the EDUs of DUC 2004. The selected relations are Elaboration, Explanation, Result, Condition, Contrast, Evaluation, and Background.

If we analyse the results with chapter 4, it is evident that there is a 7.24% gain in recall of Rouge 1 score compared with Grapharizer, 3.54% in precision score, and 4.76% in F-Score. Similarly, in Rouge 2, the gain in recall over that of Grapharizer is 5.20%, precision 2.32% and that of F-score is 3.21 %. In Rouge SU the recall gain is 8.53%, that of precision is 5.92% and increase found in F-score is 7.34%. Rouge SU4 indicators too are showing increment overall, for example the gain in recall is 4.84%, precision 2.05%, and F-score 2.88%. Rouge W also gives slightly better results compared to Grapharizer by an increase of over 1% in all the factors individually. However, the performance of Grapharizer is better than RSTSummarizer in Rouge L by a slight margin.

Proposed RSTSummarizer gives best scores in recall factor to a great margin. Which means that the number of n-gram (in case of Rouge -N) or skip grams (in case of Rouge-S) or longest common subsequence (in case of Rouge-L) in the system summaries are found in good number when we compare it with the reference summary and its total tokens.

We have also performed user evaluation of the RSTSummarizer in order to see the user's response on the different quality parameters regarding the summary. The results are presented in figure 5.1.

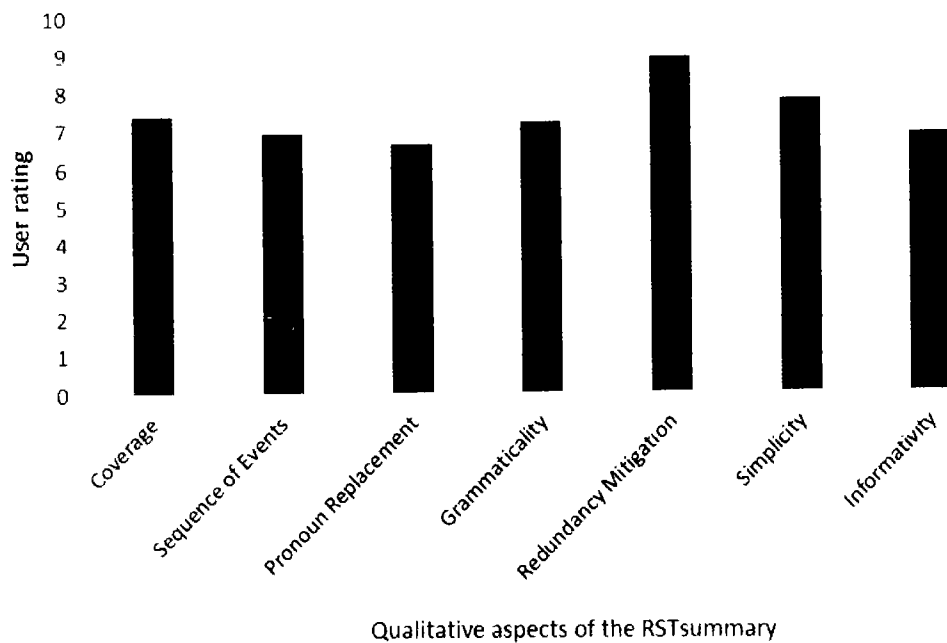


Figure 5.1: User Evaluation of RSTSummarizer

The summary's quality was assessed through a questionnaire using a Likert scale. The compiled responses showed that 69% of the evaluators found the summary informative and hence representative. It received a rating of 74% for its comprehensive coverage and diversity in capturing events from the dataset. The use of pronoun replacement was approved by 66% of the evaluators. Minimizing redundancy earned the summary a score of 89%. The evaluators ranked the summary at 70% for maintaining the sequence of events similar to the dataset. Additionally, the summary scored 78% for bringing simplicity through synonyms and MWE mapping. User Evaluation revealed that about 90% users consider the RSTSummary as mitigating the redundancy which is one of the biggest challenges of EMDS systems.

In chapter 8, we will present the results of RSTSummarizer with other summarization techniques like graph method, term based method, ontology based method, closed pattern based method, as well as different machine learning approaches so that the best approach towards extractive MDS could be identified.

Chapter 6

Implementation of Baselines

6 Introduction

ATS has proven to be an effective tool for to handle the information overload problem the Internet user is facing today. There are number of techniques created in order to serve this purpose.

In this chapter, we discuss and compare the different techniques of automatic text summarization. The techniques chosen are term-based method, ontology-based method, and a closed-pattern method. The techniques are implemented in the same settings and compared with Grapharizer (discussed in chapter 4).

6.1 Term-based method

For term weight calculation, unsupervised, extractive, and generic approaches typically use the bag-of-words model, also known as term-based methods, which frequently use the TF*ISF weighting model and various extended schemes. Term-based methods and its categories are discussed in detail in chapter 2.

For term weight calculation, the proven advantages of a term-based approach include efficiency and maturity. However, the primary flaw is that it solely considers single-word importance, ignoring polysemy and synonymy issues. **Polysemy** refers to many meanings. Polysemy is the ability of a sign or a group of signs to have many connected meanings. A polysemy is a word or phrase that has multiple meanings that are all related. For example, box (a type of tree, a container, a seating area, and to fight with fists), and wood (A piece of a tree, A geographical area with many trees). **Synonymy** is a term that refers to words that have the same or similar meanings. Synonym words are referred to as synonymous, and the state of being a synonym is referred to as synonymy. Synonyms can be nouns, verbs, adjectives, adverbs, or prepositions. For instance, begin/start, between/middle, come/arrive, and so on [121].

6.2 Ontology-based method

To create summaries, ontology-based approaches were utilised. Ontologies have been used to:

- i) identify the ideas that are most relevant to a query or most suitable for performing query expansion, and
- (ii) represent the context of summaries in a range of disciplines, like business, crisis management, etc.

Baralis et al. [38] introduced Yago-based summarization, an ontology-based technique that used Wikipedia to link words to non-ambiguous ontological concepts termed entities. Yago is an acronym that stands for Yet-Another-Good-Ontology. Yago-based summarization chooses document sentences based on the entities that have already been assigned. Ontology-based techniques are limited for certain application domains. The meanings of words are taken into account in ontology-based techniques. They can only use ontologies in a few selected application domains, and they can't get the semantic meanings of terms that don't exist in the ontology. However, ontology development is generally excessively expensive and building an ontology takes a lot of time and work.

6.3 Close patterns-based method

Many association mining technologies, including as association rule mining, frequent itemset mining, sequential pattern mining, closed pattern mining, and maximum pattern mining, have been presented in the last decade for a variety of purposes [113]. Sequential patterns come in a variety of shapes and sizes, including frequent patterns, closed patterns, etc., [114], [115]. Compared to frequent patterns, closed patterns are more information-rich and compact without compromising any information; hence, closed patterns are employed for weight calculation. Pattern-based summarization is a method for multi-document summarization that takes into account both content coverage and non-redundancy simultaneously.

In comparison to term-based techniques, it has good statistical qualities and collects more meaningful information. In comparison to ontology-based techniques, the closed pattern multi-document summarization captures informative terms from the documents. External resources, (lexical knowledgebase) are not used in this strategy. It only uses information from a document collection to build the summary.

The core problem with MDS emerged from the gathering of numerous sources from which the data was collected, which results in information redundancy, unlike SDS. Furthermore, presenting the extracted data in a coherent language to make a coherent summary is a difficult process [116].

Text summarization is primarily concerned with two issues: one, reducing repetition, and the other, providing more information in fewer words. Summary sentences are ones that can stand in for parts of a complete article. The amount of material that is repeated should be decreased, while the primary information should be maintained.

Information included in text is frequently duplicated, therefore developing a strategy to minimise duplication is critical. Different words are frequently employed in the text to describe the same object. As a result, simplistic similarity metrics between words are unable to accurately capture similarities in content. Another problem is finding substantial differences between documents and covering more informative content [44].

Multiple documents are given as input, and a summary is to be constructed from those documents. The steps in the pattern-based technique are as follows. To begin, a corpus is mined for all closed sequential patterns, after which the method represents all phrases using these closed patterns. Using the distribution of closed patterns to calculate pattern weights the sentence-representation model covers the major content of the document collection. Finally, the sentences that are the most informative and non-redundant are chosen [117].

6.3.1 Generating Closed Patterns

The first stage is to mine a corpus for all closed sequential patterns. A pattern is a set of ordered items that appear in a series of sentences frequently. The terms with a high frequency in the dataset are represented by closed patterns. Closed patterns are concise and informative, without sacrificing any information than frequent patterns.

6.3.2 Sentence-Representation

This stage uses a closed sequential pattern method to extract all closed sequential patterns from the document collection. Each sentence is represented by its terms and their respective weights, which are calculated by adding the weights of the sentence's covering closed patterns.

6.3.3 Sentence-Ranking

Sentences are sorted according to the preceding sentence-representation in order to select the most relevant and meaningful sentences for the summary.

6.3.4 Sentence-Selection

The sentences with more closed patterns of high weights, and more different than the other selected sentences chosen into the summary. This process continues till the length limit is reached. After performing all these processes, the required summary is generated as output.

6.4 Experiment and discussion

We have implemented all the three methods stated above on same settings in Python at DUC 2004, and compared the results with Grapharizer, the technique discussed in chapter 4, and with RSTSummarizer, the technique discussed in chapter 5. The results are presented in table 6.1. why it was necessary to re-implement the techniques instead of using their results stated in paper? The performance of algorithms varies when they are executed on different machines with varying specifications. Similarly, the libraries used can also contribute to the efficiency of the algorithms. Since we have kept the settings similar for all the methods, therefore the results are exactly showing the comparison in a very minute and clear way.

Table 6.1: Comparison of Grapharizer with State-of-The-Art Systems

ROUGE evaluation		Methods			
	METRICES	Term method	YAGO Ontology	Close Patterns	Grapharizer
<i>Rouge 1</i>	Recall	0.37390	0.41757	0.50852	0.46136
	Precision	0.15964	0.19432	0.26866	0.21030
	F-Score	0.22375	0.26522	0.35157	0.28891
<i>Rouge 2</i>	Recall	0.06527	0.09250	0.15571	0.09822
	Precision	0.02851	0.04384	0.08564	0.04203
	F-Score	0.03968	0.05949	0.11050	0.05886
<i>Rouge L</i>	Recall	0.28360	0.27042	0.39063	0.17050
	Precision	0.17917	0.17083	0.28333	0.08979
	F-Score	0.21960	0.20939	0.32844	0.11763
<i>Rouge W</i>	Recall	0.08782	0.11257	0.12043	0.10299
	Precision	0.07168	0.09557	0.11172	0.08422
	F-Score	0.07893	0.10338	0.11591	0.09238
<i>ROUGE-SU4</i>	Recall	0.12943	0.15407	0.20314	0.17506
	Precision	0.05840	0.07461	0.11741	0.07078
	F-Score	0.08048	0.10054	0.14881	0.10080

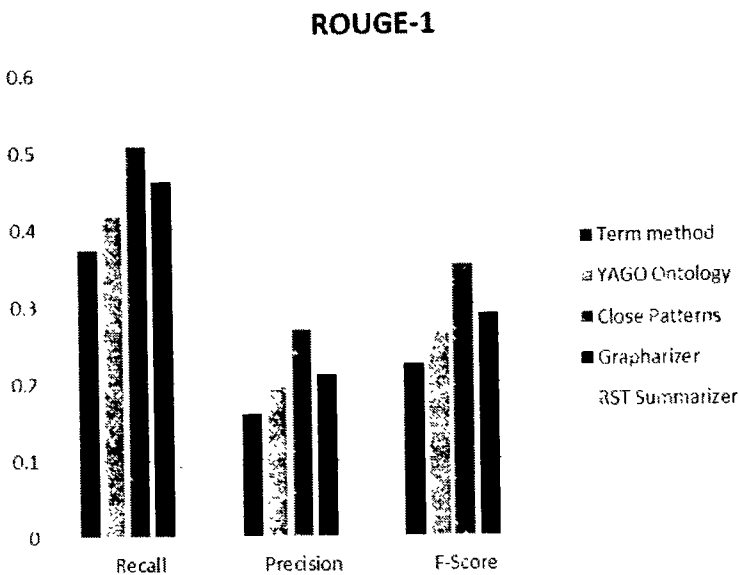


Figure 6.1: Rouge 1 comparison of Grapharizer with SOTA methods

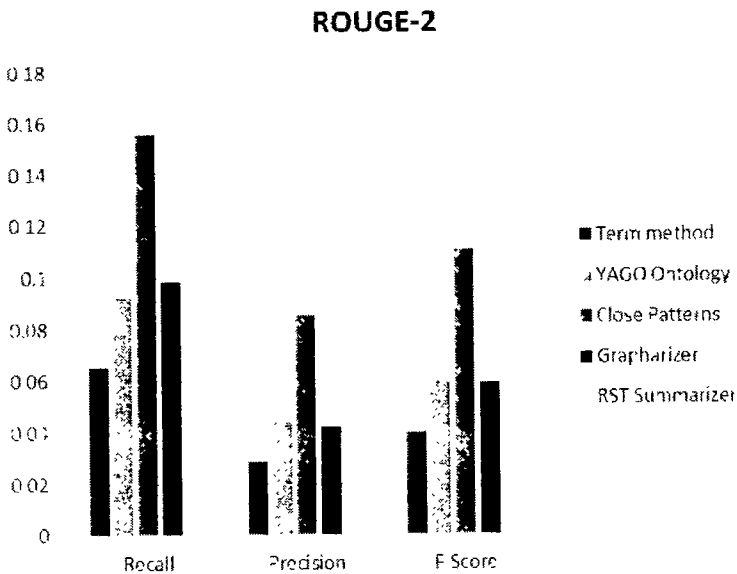


Figure 6.2: Rouge 2 comparison of Grapharizer with SOTA methods

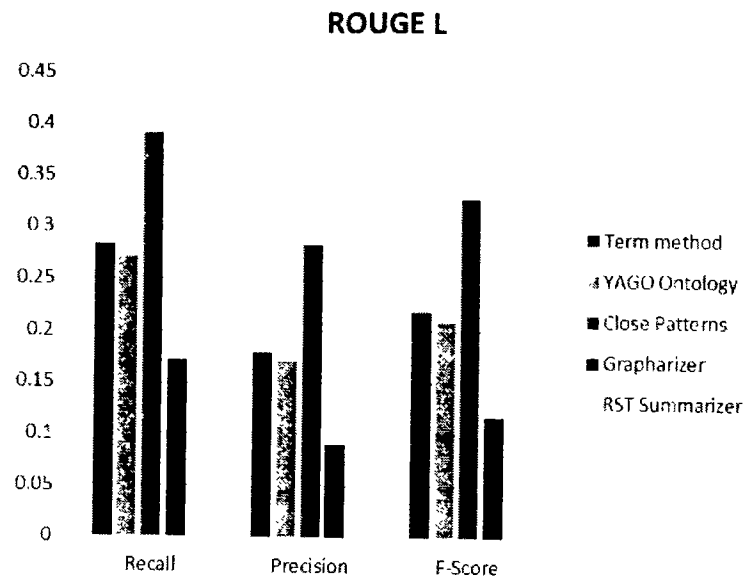


Figure 6.3: Rouge L comparison of Grapharizer with SOTA methods

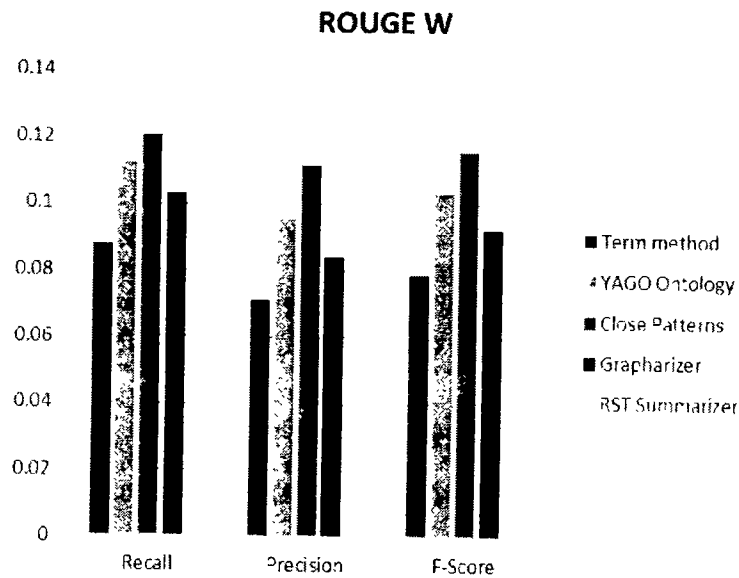


Figure 6.4: Rouge W comparison of Grapharizer with SOTA methods

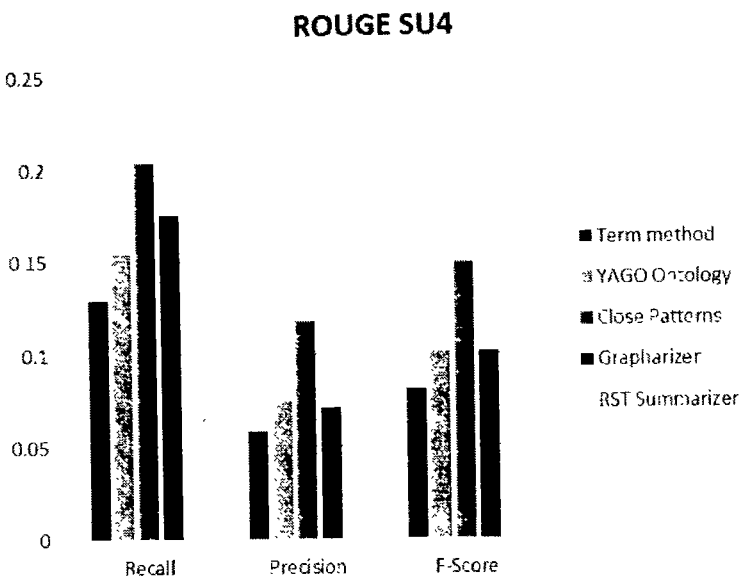


Figure 6.5: Rouge SU comparison of Grapharizer with SOTA methods

As shown in the table 6.1, and the figure 6.1 – figure 6.5, the results generated by closed-pattern based method performed better than the rest of the techniques in almost all metrics of Rouge evaluation in terms of precision and f-score. RSTsummarizer, however, performed better in recall measure of Rouge 1 and Rouge SU4. The results of RSTSummarizer infer that the unigrams in Rouge 1 and the skip-grams of SU4 exist in the system generated summaries are in large number, that is why when divided by the total tokens of the reference summary, the score remained the highest. This also indicates towards the accuracy of the representativeness module of the RSTSummarizer that it picked the exact saliant words as were picked up in the reference summaries.

Chapter 7

Machine Learning

7 Introduction

Sentence extraction is important in extractive MDS. By controlling the sentence features, we can find out the salience scores. When we have more input sentences, then it becomes hard to focus on dealing with sentence features individually. That is when we need machine learning in ATS. With vast feature set, most of the machine learning methods over-train the training data [109].

7.1 Support Vector Machines

SVM is a supervised learning approach for dividing data into dimensional hyperplane based on binary class [110]. The important vs. non-important sentences correspond to the positive and negative examples. Margin is the distance between positive and negative examples. SVMs attempt to achieve the maximum margin between classes in order to discriminate between negative and positive examples [111]. Therefore, it aims to determine optimal hyperplane [109]. The points that are closest to the hyperplane are known as support vectors. These data points will be used to define a separation line. Margin is the separation between the hyperplane and the observations (support vectors) that are closest to the hyperplane. Large margins are regarded as good margins in SVM. As we know linearly separating the data is not always easily possible, therefore misclassification can occur. Slack variables are used to correct the errors. We have adapted the working of Tzouridis et al. [54] for implementation of SVM.

7.2 Artificial Neural Networks

Artificial neural networks (ANNs) are self-learning systems made up of interconnected neurons and nodes with input and output. They are utilised to find or detect solutions or features that would be difficult to find using traditional programming [110]. The Feed-Forward Neural network is designed following the mechanism of Sinha et al. [122] where the features are fed to the input layer, given to the hidden layer for processing and output is given to the output layer. For the errors in classification, the output is sent back to the input layer for corrections.

7.3 Multivariate Linear Regression

Another popular machine learning algorithm is multivariate linear regression. A situation with more than one independent variable arrives when linear regression fails to work, multivariate regression is used. Multiple variables or features are present in real-world data, therefore Multivariate regression is required for better analysis. Multivariate Regression is a supervised ML approach that analyses numerous data variables with one dependent variable and several

independent variables. We try to anticipate the outcome based on the number of independent variables. The most useful aspect of multivariate regression is that it aids in the understanding of relationships among variables in a dataset [110]. This makes it easier to comprehend the relationship between dependent and independent variables.

To evaluate the performance of the machine learning based methods, we have applied these methods along-with the Grapharizer technique, developed as module 1 (discussed in chapter 4) of the thesis.

As shown in figure 3.4, in order to train the model, we have to find k-short paths from the graph. We have employed the Yen's algorithm for this purpose. The features used are given below:

1. **Joint frequency** $\phi_1(w, w') = e/n$ = frequency of edge/no. of vertices in graph.
2. **Maximal word frequency** $\phi_2(w, w') = \max \{w/n, w'/n\} = \max \{\text{freq. of } v/\text{no. of vertices}, \text{freq. of } v'/\text{no. of vertices}\}$
3. **Lexical relevance** $\phi_3(w, w') = (2/n) \cdot (w \cdot w' / (w + w'))$
4. **Normalized PMI** $\phi_4(w, w') = (\log(e/w \cdot w')) / (-\log e/n) = (\log e/w \cdot w') - \text{joint frequency}$
5. **Average location** of the phrase or group of words in the input sentence

Along with these features, additional features are included like:

Lexical: Indicator features on non-stopwords that appear more than five times in training set and its similar part-of-speech features. First word, last word, preceding word, and following words in each textual unit are to be considered, along with the index of sentence having the textual unit in the document.

Structural: It includes the conjunctions of position of textual unit in the document, its length, length of the corresponding sentence, index of the paragraph it occurs in, whether it is start of the new paragraph.

Centrality: It's about centrality of content, word counts coupled with sentence index in multiple documents. Also, to include in features the frequency of entity mention in a sentence in the rest of the documents, the count of mentioned entities in a sentence, surface properties of entities (type, length).

Pronoun replacement: Such as the length of pronoun replacement, its sentence distance from the current mention, its type (nominal or proper), identity of the pronoun being replaced.

Experimental Set-up:

Our experiments were conducted on two datasets: DUC 2004 and Recent News Articles. DUC 2004 comprises fifty additional datasets, each containing ten documents related to a news topic, along with four reference summaries. To assess the performance of our proposed system, we employed cross-validation. Specifically, we divided the datasets into training and testing sets, with twenty-five datasets allocated for each phase. The even-numbered datasets were used for training, while the odd-numbered ones were used for testing. The summary length in this dataset was set to 150 words, with four summaries generated by human experts.

Regarding the Recent News Articles dataset, it consisted of twenty-five additional datasets, each containing ten documents on a specific topic. Like DUC 2004, we employed cross-validation to evaluate the system's performance. For each iteration, we selected thirteen datasets for training and twelve datasets for testing. In this dataset, the summary length was set to 200 words, and the summaries were generated using ChatGPT. Similar to DUC 2004, we have provided four reference summaries generated by ChatGPT for assessment of system summaries.

Then SVM, MLR and ANN are applied to test the results generated by these techniques using Grapharizer. The results are presented in table 7.1.

Table 7.1: Comparison of Grapharizer with Machine Learning Methods					
ROUGE evaluation		Methods			
	METRICES	Grapharizer	SVMGrapharizer	ANNGrapharizer	MLRGrapharizer
Rouge 1	Recall	0.46136	0.48746	0.49161	0.49668
	Precision	0.21030	0.22898	0.23858	0.23136
	F-Score	0.28891	0.31159	0.32126	0.31567
Rouge 2	Recall	0.09822	0.13379	0.11064	0.11072
	Precision	0.04203	0.05942	0.05046	0.04889
	F-Score	0.05886	0.08229	0.06931	0.06783
Rouge L	Recall	0.17050	0.21428	0.20479	0.21532
	Precision	0.08979	0.12311	0.12016	0.12037
	F-Score	0.11763	0.15637	0.15145	0.15442

<i>Rouge W</i>	Recall	0.10299	0.08244	0.08647	0.07991
	Precision	0.08422	0.07031	0.07543	0.06754
	F-Score	0.09238	0.07589	0.08058	0.07321
<i>Rouge SU</i>	Recall	0.18137	0.15686	0.16176	0.16176
	Precision	0.06751	0.07207	0.07603	0.07366
	F-Score	0.09840	0.09876	0.10344	0.10122
<i>ROUGE-SU4</i>	Recall	0.17506	0.19238	0.18077	0.18128
	Precision	0.07078	0.08049	0.07765	0.07531
	F-Score	0.10080	0.11349	0.10863	0.10641

It is evident from the data presented in the table above that except for Rouge W and the Recall measure of Rouge SU, machine learning based algorithms performed much better than the Grapharizer method. SVMGrapharizer improved the result of Rouge 1 recall by 2.61%, precision by 1.87%, and the f-score by 2.27%. ANNGrapharizer improved recall in Rouge 1 by 3.03%, precision by 2.83%, and the f-score by 3.24%. Similarly, MLRGrapharizer improved the recall measure of Rouge 1 by 3.53%, precision by 2.11% and f-score by 2.68%.

SVMGrapharizer emerged as leading scorer in generating summaries closer to the reference summaries in most of the Rouge measures, specifically Rouge 2, Rouge L, and Rouge SU4. ANNGrapharizer performed better in Rouge 1 (precision, f-score) and Rouge SU (precision and f-score). The notable performance of MLRGrapharizer is reported in Rouge 1 recall, and Rouge L recall only.

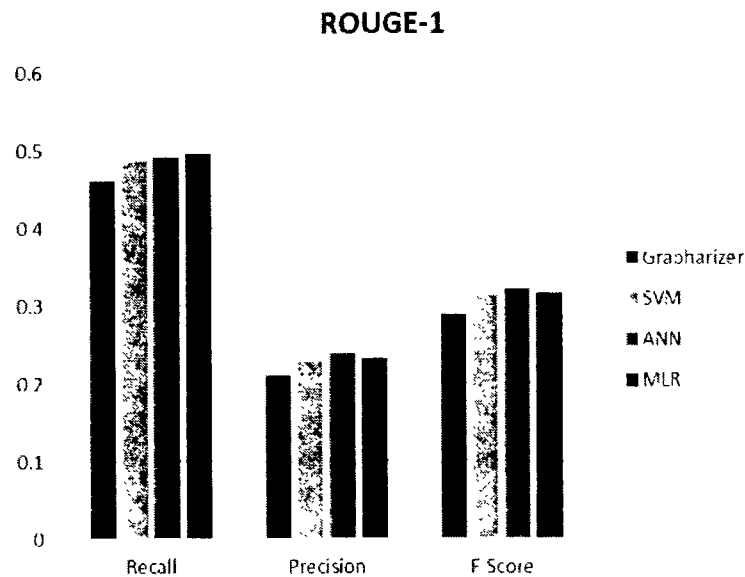


Figure 7.1: Rouge 1 Comparison of Grapharizer with ML variants

In Rouge 1, as shown in figure 7.1, the Recall based performance of MLRGrapharizer topped the rest of the methods, ANNGrapharizer scored second best and SVMGrapharizer is third on the performance. ANNGrapharizer improved the Precision score, however that not only impacted in best score for ANNGrapharizer but also resulted in best F-score results for Rouge 1.

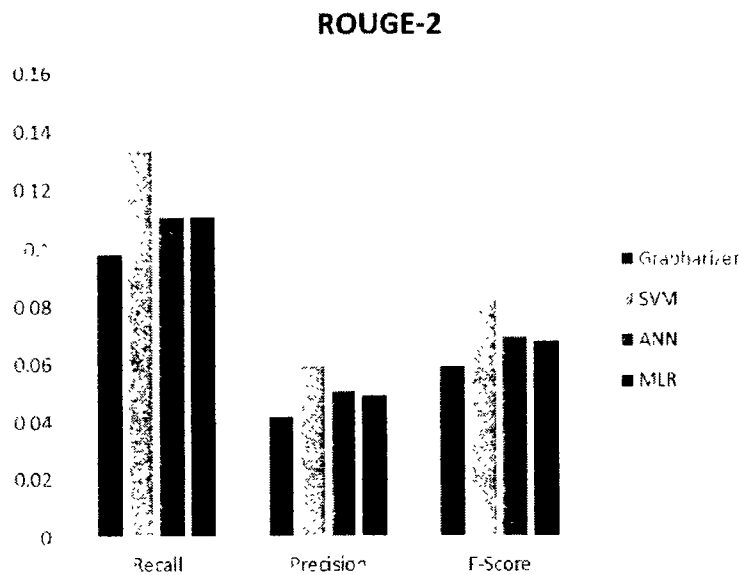


Figure 7.2: Rouge 2 Comparison of Grapharizer with ML variants

Figure 7.2 shows that SVMGrapharizer scored much better than the rest of the methods in Recall measure of Rouge 2. Performance of MLRGrapharizer and ANNGrapharizer remains identical. Almost similar trend is observed in Precision and F-score, with slight edge of ANNGrapharizer over MLRGrapharizer, however.

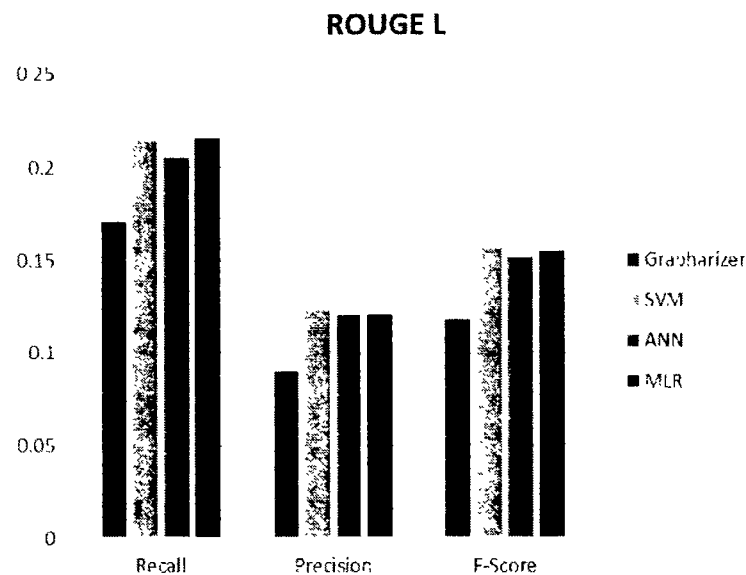


Figure 7.3: Rouge L Comparison of Grapharizer with ML variants

Figure 7.3 shows a close competition between MLRGrapharizer and SVMGrapharizer. MLRGrapharizer performed slightly better for Recall in Rouge L, while in Precision and F-score, the performance of SVMGrapharizer is reported to be better.

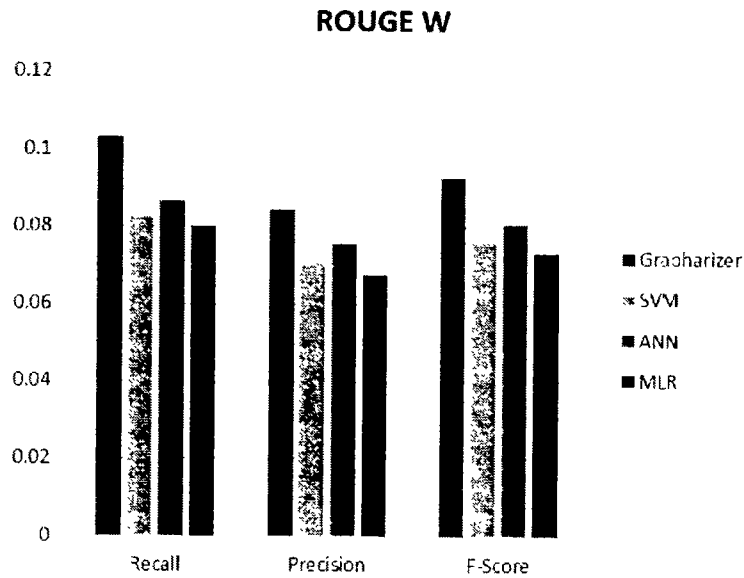


Figure 7.4: Rouge W Comparison of Grapharizer with ML variants

Figure 7.4 shows the results for Rouge W that favours weighted longest consecutive subsequence and it is surprising to notice that Grapharizer alone is outperforming its other ML based variants in almost all the 3 measures, namely precision, recall and F-score, for all the ML techniques, be it SVM, ANN or MLR.

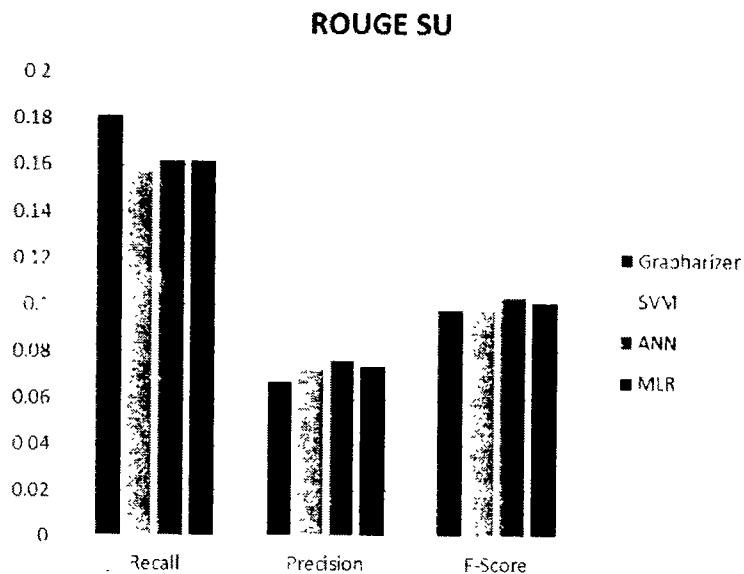


Figure 7.5: Rouge SU Comparison of Grapharizer with ML variants

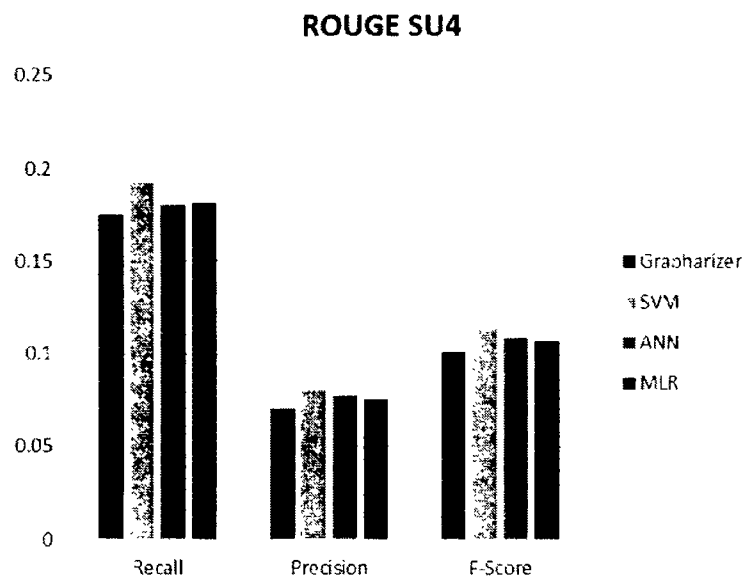


Figure 7.6: Rouge SU4 Comparison of Grapharizer with ML variants

Figure 7.6 Rouge SU4 is similar to Rouge-N except that it includes the bigrams of skip distance of even four words. In that measure, the Performance of SVMGrapharizer is better in precision, recall and f-score.

Chapter 8

Results & Comparative Analysis

8 Results and Discussion

In this thesis, we have designed a framework for EMDS, where we have devised a graph theory based system and an RST theory based system. When then tested these systems with machine learning based techniques to gauge the effect of machine learning in EMDS. Based on the results of the survey to explore current evidence in EMDS [81], we explored that majorly the solutions to EMDS problems are comprised of term based, ontology based, close pattern based, graph based and RST based systems. Therefore, we have also developed term based, ontology based and closed pattern based systems for better comparison and evaluation of our solution.

8.1 Comparison of Results with related work

We have devised two algorithms in this thesis. One related to graph theory to provide best solution for EMDS, and another related to Rhetoric Structure Theory based solution for EMDS. We have also implemented baselines from popular EMDS solutions and compared the devised solutions with them. We have also implemented popular Machine Learning algorithms for EMDS and a detailed quantitative analysis was carried out using ROUGE 2.0 for evaluation. The results of the detailed comparison are discussed in subsequent sections. The summaries generated are presented in Appendix B.

8.2 Quantitative Analysis

The results of thesis are presented in table 8.1. Since we have used ROUGE 2.0 for the evaluation of our results, therefore we need to closely see the implication of its individual results. We have normally used Rouge 1, Rouge 2, Rouge W, Rouge L, Rouge SU, and Rouge SU\$ for the precision, recall and F-score values. Now let's understand what these results actually interpret. Rouge-N means the matching N-grams between the summary generated by the system and the reference summaries present in DUC 2004. N-gram is the collection of tokens or words. For Rouge 1, the size of the n-gram is 1 gram, i.e., unigram, so we must check for the matching words in a word-by-word fashion between the two summaries, one generated by our system, and the other provided by the dataset. In a sentence "Cambodian prime minister Hun Sen rejects demands of opposition parties for talks in Beijing" the unigrams are "Cambodian", "prime", "minister", "Hun", "Sen", "rejects", "demands", "of", "opposition", "parties", "for", "talks", "in", and "Beijing". The examples of bigram, i.e., 2 grams are "Cambodian prime", "prime minister", "minister Hun", "Hun Sen" and so on.

Table 8.1: Rouge scores of the SOTA systems, devised systems, and ML algorithms

Rouge Metrics		Grapharizer	RST Summarizer	Term method	YAGO Ontology	Close Patterns	SVM Grapharizer	ANN Grapharizer	MLR Grapharizer
Rouge 1	Recall	0.46136	0.53374	0.3739	0.41757	0.50852	0.48746	0.49161	0.49668
	Precision	0.2103	0.24567	0.15964	0.19432	0.26866	0.22898	0.23858	0.23136
	F-Score	0.28891	0.33647	0.22375	0.26522	0.35157	0.31159	0.32126	0.31567
Rouge 2	Recall	0.09822	0.15025	0.06527	0.0925	0.15571	0.13379	0.11064	0.11072
	Precision	0.04203	0.06522	0.02851	0.04384	0.08564	0.05942	0.05046	0.04889
	F-Score	0.05886	0.09095	0.03968	0.05949	0.1105	0.08229	0.06931	0.06783
Rouge L	Recall	0.1705	0.15732	0.2836	0.27042	0.39063	0.21428	0.20479	0.21532
	Precision	0.08979	0.08217	0.17917	0.17083	0.28333	0.12311	0.12016	0.12037
	F-Score	0.11763	0.10795	0.2196	0.20939	0.32844	0.15637	0.15145	0.15442
Rouge W	Recall	0.10299	0.11372	0.08782	0.11257	0.12043	0.08244	0.08647	0.07991
	Precision	0.08422	0.09742	0.07168	0.09557	0.11172	0.07031	0.07543	0.06754
	F-Score	0.09238	0.10494	0.07893	0.10338	0.11591	0.07389	0.08058	0.07321
Rouge SU	Recall	0.18137	0.26667				0.15686	0.16176	0.16176
	Precision	0.06751	0.12669				0.07207	0.07603	0.07366
	F-Score	0.0984	0.17178				0.09876	0.10344	0.10122
ROUGE-SU4	Recall	0.17506	0.22342	0.12943	0.15407	0.20314	0.19238	0.18077	0.18128
	Precision	0.07078	0.09127	0.05841	0.07461	0.11741	0.08049	0.07765	0.07531
	F-Score	0.10081	0.12959	0.08048	0.10054	0.14881	0.11349	0.10863	0.10641

Each metric of Rouge is providing the users three factors to inform about the quality of the generated summary in more depth. These factors are Recall, Precision, and the F-score. Recall counts the overlapping n-grams in both the summaries then divides the score with total n-grams in the reference summary. If the system summary contains more words that are present in the reference summary, the recall score will be higher.

Precision factor is little different from recall, as it counts the common n-grams of both the summary but divides with the total n-gram count of the system summary instead.

When we compared Grapharizer with the baseline SOTA systems, the recall metric of Rouge-1 show improvement by 8.75%, precision with 5.07%, and f-score improved by 6.52% against term based system. The improvement was recorded to be 4.38% in recall against ontology based system, 1.6% improvement in precision was gained, while f-score increased by 2.37%. Closed patterns based system however performed better than Grapharizer in all the three metrics of Rouge-1.

For Rouge-2, Grapharizer performance increased 3.30% in recall against term based method, 1.35% in precision, and 1.92% in f-score. When compared against ontology method, our approach performed well in recall by 0.57%.

For Rouge-W, Grapharizer improved the recall metric by 1.52% margin with term method, 1.25% in precision, and 1.35% in f-score. For Rouge-SU4, the recall against term based improved 4.56%, precision 1.24%, and f-score 2.03% improved. The recall improvement was noted to be 2.10% against ontology based system, and f-score slightly improved by 0.03%.

The second system we proposed, the RSTSummarizer, was also checked for performance against these baselines. In Rouge-1, recall improvement was recorded to be 15.98% against term based, 8.60% in precision, and 11.27% in r-score. When compared with ontology based system summary, it performed 11.60% better in recall, 5.14% improved in precision and 7.13% for f-score. It improved the recall against closed patterns by a margin of 2.52%.

Rouge-2 also resulted in favor of RSTSummarizer. Against term based, the recall improved 8.5%, precision with 3.67%, and f-score with 5.13%. Against ontology based, the recall improved 5.78%, precision with 2.14%, and f-score 3.15%.

Recall of Rouge-W improved against term based with 2.59%, precision 2.57% and f-score 2.60%. Against ontology summary, the recall improved 0.12%, precision 0.19% and f-score 0.16% improved for RSTSummarizer. Similarly, improvements were noted in Rouge-SU4

against the baselines. Against the term summary, recall improved 9.40%, precision improved 3.29% and f-score 4.91%. Against ontology based, recall improved with 6.94%, precision with 1.67% and f-score with 2.91%. Moreover, the recall was improved against closed patterns with 2.03% margin.

We have implemented three ML methods to provide summarization solutions. In this section we discuss the Rouge scoring of ML methods against the baseline SOTA summarizers.

We begin the discussion with SVMGrapharizer. The performance of SVM improved against the baselines in Rouge-1. Recall against term summary increased by 11.36%, precision by 6.93% and f-score by 8.78%. When compared with ontology summary, SVM performed well by recall increment of 6.99%, precision by 3.47%, and f-score 4.64%.

Rouge-2 recall against term based improved by 6.85%, precision by 3.09% and f-score by 4.26%. When tested against ontology summarizer, the improvement was 4.13%, precision was 1.56%, and f-score 2.28%. Rouge-SU4 recall against term summary improved 6.30%, precision 2.21%, f-score 3.30%. Score of recall against ontology summary was improved by 3.38%, precision 0.59%, and f-score 1.30%.

Then we discuss about ANN and its performance against the baselines. Rouge-1 recall against term based improved by 11.77%, precision 7.89%, f-score 9.75%. When compared with ontology summary, the ANN recall improved 7.40%, precision 4.43% and f-score 5.60%.

Rouge-2 recall of ANNGrapharizer gained 4.54% score against term summarizer, 2.20% precision, and 2.96% f-score. Recall remained 1.81% increased when tested against ontology summarizer, precision 0.66% and f-score 0.98% improved. Rouge-L reported slight improvement in precision against term summary by 0.38% and f-score by 0.17%. Rouge-SU4 improved against term summary by a margin of 5.13% recall, precision by 1.92% and f-score by 2.82%. Ontology summary also did not perform well against ANNGrapharizer. The recall improved 2.67%, precision by 0.30%, and f-score 0.81% for ANN.

The third ML technique adapted for Grapharizer was MLR. Rouge-1 recall against term summary was improved by 12.28%, precision by 7.17%, and f-score by 9.19%. MLR was tested against Ontology summary and the improvement was 7.91% in recall, 3.70% in precision, and 5.05% in f-score.

Rouge-2 recall against term summary improved 4.55%, precision 2.04%, f-score 2.82%. Ontology summary was tested and MLR improved 1.82% in recall, 0.51% in precision, and

0.83% in f-score. Rouge-SU4 recall improved against term summary by 5.19%, precision 1.69%, f-score 2.59%. MLR worked well against ontology summary with recall improved 2.72%, precision 0.07%, and f-score 0.59%.

Table 8.1 contains some boldfaced values, which represents the top score in that specific Rouge score, and the yellow highlighted values are the second seeded values at the same metrics. The results shown in table 8.1, and the discussion above represent very different picture of the scores than the quality of the summaries. Therefore, we decided to apply qualitative analysis for having a true picture of the results.

8.3 Qualitative analysis

User evaluation of summaries is important for assessing quality of the summaries [123]. Therefore, we have performed the user evaluation at multiple stages to assess the Grapharizer and RSTSummrizer's procedures.

To begin with, we evaluated the effectiveness of synonym mapping by the users. For that, we compiled a file of some random twenty words, and their synonyms assigned from thesaurus.com using our implemented function named `cross()`. We evaluated the accuracy through expert evaluation. Three experts were requested to validate the accuracy of synonyms assigned by `cross()`. Thesaurus.com color codes the synonyms in three levels. The most relevant synonyms of words are represented in red color on the website, slightly far in meanings are coded in orange color, and remotely synonymous words are shown in yellow color. Keeping this in focus, the words and synonyms were provided to the experts on Likert scale, 1 to rate no synonymy, and 10 to show high synonymy between the two words. The score given by expert 1 for accuracy of synonyms was 92%, and that by expert 2 was 89%. Expert 3 scored the provided words and their synonyms as 84% accurate. On average, 88% accuracy was obtained for the synonyms by our `cross()`. With this score, we applied the function on DUC 2004 and Recent News Articles for our Grapharizer.

Afterwards, we have evaluated the summary generated by Grapharizer by the users and hired twenty-five evaluators to assess the summary after reading the relevant news article provided in the dataset. A questionnaire was provided regarding the summary quality. They marked their answers on the provided Likert scale. The responses were then compiled and 84% score was given by evaluators to the informativity and representativeness of the summary. Pronoun replacement was approved with a score of 76% by the human evaluators. 88% score was given for minimizing redundancy in the summary. However, when the evaluators were asked to

compare the summary with some human generated summary, they rated the Grapharizer summary to 55% in that case.

Finally, the summary produced by RSTSummarizer was assessed by a team of sixty evaluators who read the relevant news article provided in the dataset. A questionnaire was given to gauge the quality of the summary, and the evaluators used a Likert scale to provide their responses. The compiled responses revealed that 69% of the evaluators considered the summary to be informative and representative. Additionally, the summary was rated 74% for its comprehensive coverage of events in the dataset and its diversity. The use of pronoun replacement received a 66% approval rating from the evaluators. The summary was given an 89% score for minimizing redundancy. Evaluators ranked the summary at 70% for maintaining the sequence of events similar to the dataset. Furthermore, the evaluators assessed that the summary brought simplicity through synonyms and MWE mapping, giving it a score of 78%.

8.4 Statistical Testing

We conducted a non-parametric test, i.e., ANOVA, to compare Grapharizer with state-of-the-art techniques in terms of ROUGE scores. The result, with a p-value of 0.91 supporting the alternative hypothesis H1, indicates that there were no significant differences between the samples. In other words, we can conclude that Grapharizer gives comparable results to SOTA MDS techniques.

8.5 Answers to Research Questions

At the end, we will answer the research question that we posed at the beginning of the thesis.

Question 1: What research evidence has been reported in the literature on extractive multi-document summarization?

Answer: we have conducted a thorough literature review [81] that mentions research evidence in the field of extractive MDS.

Question 2: How to devise and implement an EMDS system using graph theory?

Answer: We have conducted an experiment (mentioned in chapter 4) that sheds the light on the improvements brought by the graph-based method in the field of extractive MDS [124].

Question 3: How to devise and implement learning based EMDS system using graph theory?

Answer: We have performed an experiment that mentions the improvements by the machine learning techniques to extractive MDS [124]. We have discussed it in chapter 7.

Question 4: How to design and implement RST based EMDS system?

Answer: In chapter 5, we have discussed another experiment that we have conducted to show the improvements of RST based summarizers to the extractive multi-document summarization.

Question 5: How to implement SOTA baseline systems for the purpose of comparison?

5.1: How to implement term based EMDS system

5.2: How to implement closed pattern based EMDS system

5.3: How to implement ontology based EMDS system

Answer: Experiments conducted that are stated in chapter 6 answers this question and its sub-parts.

Question 6: Which of the following techniques perform better in comparison with the SOTA EMDS systems (i.e., term based, ontology based, close pattern based) w.r.t. qualitative (informativity, representativeness, grammaticality, diversity, redundancy mitigation etc.) and quantitative (ROUGE evaluation) parameters?

- a. Graph based EMDS system.
- b. Machine learning based EMDS system.
- c. RST based EMDS system.

Answer: Experiments conducted that are stated in chapter 6, 7 and 8 answer this question.

Chapter 9

Conclusion & Future Work

9 Conclusion and Future Work

This chapter concludes the thesis and highlights the future work that needs to be done.

9.1 Conclusion

This thesis presents in chapter 2 a recent survey of previous work on using extractive techniques for multi-document summarization. It would provide a perfect starting point for the researchers to contribute to the field of multi-document summarization. Extractive techniques can be divided into Term-based methods, Rhetoric Structure Theory-based methods, graph-based methods, and several other variations of the most standard methods. The working of these techniques is individually explained, and then a thorough discussion on the important studies conducted is carried out. We discussed the strengths and weaknesses of each method under different training conditions. We also presented the most commonly available datasets that are used to evaluate and compare new summarization techniques. In the end, we present the evaluation matrices. We have discussed the strengths and weaknesses of different techniques and pointed out future directions for newcomers in research. Table 1-5 in chapter 2 can be especially beneficial for the researchers who wish to find research problems to kick start their research process. Several studies can be considered for improvements. For example, the techniques of [4], [35], [43], [55], [64], [72] can be rigorously tested on datasets like DUC, TAC, NYT, and other benchmark datasets, discussed in section 3.

Similarly, the work of [57] can show significant improvements if pre-processing is enhanced with the step of stemming. It is worth mentioning that polysemy and synonymy are repeatedly reported as an open issue in MDS literature. However, it has not attracted much attention from researchers so far. The interested researchers can kick start their endeavor by handling it in many studies, for instance, [53], [63].

One of the observations of this study is that external resources, like WordNet, are frequently used for synonym mapping. For the synonym mapping, these studies can be replicated using word embedding techniques [87], e.g., Word2Vec, GloVe, etc. Moreover, in [47], all important events were not included in the intermediate summary. In [81], the sentence extraction from multiple topics did not work well, while in [59], the progress deteriorates with varying lengths of summaries. In study [46] they made satisfactory grammatical improvements to their summary. However, they could not improve the informativity. The researchers can improve this study [46] for informativity. The feature-set of work of [68] can be extended to improve the results; on the other hand, [56] approaches the researchers to work on improving summary

cohesion. In the end, the summarizers with positive results in SDS [42], [75] can be adapted for improved performance in MDS.

ATS work is mainly focused on the English language. The benchmark datasets and dictionaries like WordNet, Thesaurus.com, etc., supporting the English language. The research community, however, is trying to make summarization systems in their native languages too. [82] has tested its model in Spanish alongside English. [10] focused its study on the Arabic language, while [64] has focused on the Turkish language. [43] and [44] is implemented for text summarization in the Japanese language. Urdu is one of the widely spoken languages in Asia; therefore, there is a need to make an ATS system that could facilitate document summarization for the Urdu language. For that, proper language dictionaries and standard datasets with gold standard summaries must be developed [88]. This transformation can further be facilitated by employing word embedding techniques [87]. With proper provision of datasets and dictionaries, any of the discussed techniques in the survey can be adapted to other languages.

Automatic Text Summarization systems are increasingly gaining the interest of the users to obtain the concise version of the lengthy and redundant textual documents, without skipping any important piece of information. In this thesis, we presented the state-of-the-art techniques published in different studies over the last decade about EMDS. We discussed in detail the various techniques of extractive MDS, like i) ontology-based methods, ii) term-based methods (that can be further classified into clustering methods, latent-semantic methods, and non-negative matrix factorization) iii) rhetoric structure theory-based methods, and iv) the graph-based methods. We proposed and discussed the different guidelines to facilitate the new researchers in this field to make a start, emphasizing the abovementioned techniques. We critically analyzed and discussed several studies and pointed out their strengths and weaknesses. As mentioned in the review paper [81], we have identified the following open issues and tried our best to bring improvements in the research as well except for the 5th point. The open issues are as follows:

- i) **Diversity:** To increase its diversity, every topic mentioned in the document clusters should be mentioned in summary. It must not be focused upon just one of the many topics found in the document clusters.
- ii) **Redundancy:** The text summarization systems mainly suffer from the repetition of the same fragments of information in summary [86], ignoring many important points, therefore.

The need is to devise a summary in such a manner that repetition should be minimized, if not eliminated.

iii) **Informativity:** The summary must be carrying the information in a precise and concise manner. Extractive summarization involves extracting the fractions from the given documents; therefore, it mainly suffers from the lack of informativity concerns. An effective summarizer must convey the information in a compact way to the reader.

iv) **Grammaticality:** The quality of the summary suffers from grammar due to connecting the extract of the different chunks from the document set. The need is to make such a system that refines the summary for grammar at the end.

v) Urdu is among the popular languages spoken worldwide, but unfortunately; no summarizer is available for the Urdu language script. It can be a potential NLP research area to focus future research direction.

In the era of information overload, it is necessary to have access to accurate information in less time. The need is to have such concise accurate information provided against user queries that will not have redundant information, and it must contain all the aspects of the required subject.

In this thesis, we anticipated that the more data is prepared for processing in the pre-processing stage, the better would be the results. Furthermore, to avoid redundancy and to ensure the maximum coverage of all the topics of the input documents in the summary (pointed out in the points I, ii, and iii above), we introduced the intermediate step of topic modeling before the actual processing. That resulted in a concise and informative summary that performed better than the different SOTA techniques.

To maintain the poetic flow of the documents reflected in the summary, we have applied the `reverse-mwe()` that served the purpose well to handle the issue mentioned above in point iv.

In comprehensive experiments, we have applied the Grapharizer with different benchmark machine learning algorithms like SVM, ANN and MLR, and named it SVMGrapharizer, ANNGrapharizer, and MLRGrapharizer. We have tested the hypothesis that whether machine learning contributes positively to summary generation process or not. The results confirm the positive contribution of the machine learning to the summarization process. The RSTsummarizer improved the summary grammaticality as well. The cohesion is found in the summary by RSTsummarizer.

We have tested the RSTsummarizer and Grapharizer and its ML based variants against the SOTA methods like term based, ontology based and close pattern based methods. Except for the closed pattern, our developed techniques worked better than the rest of SOTA methods.

9.2 Future Work

We have conducted extensive experiments for evaluation of the techniques of summary generation. There is however still a need to apply different evaluation techniques, like BERTScores, BLEU, and manual evaluation by using crowdsourcing platforms like Amazon's Mechanical Turk, or UpWork to check the summary quality in different dimensions too.

Similarly, the results of the techniques vary with different datasets. Therefore, we would strongly encourage the researchers to test the methods devised and evaluated in this thesis over various datasets.

The closed pattern-based method may also be tested with ML techniques to check whether the results are further improved or not.

Appendix A

References

References

- [1] A. Khan, N. Salim, and Y. Jaya Kumar, "A framework for multi-document abstractive summarization based on semantic role labelling," *Appl. Soft Comput. J.*, vol. 30, pp. 737–747, 2015, doi: 10.1016/j.asoc.2015.01.070.
- [2] R. Ferreira et al., "A multi-document summarization system based on statistics and linguistic treatment," *Expert Syst. Appl.*, vol. 41, no. 13, pp. 5780–5787, 2014, doi: 10.1016/j.eswa.2014.03.023.
- [3] G. Yang, D. Wen, Kinshuk, N. S. Chen, and E. Sutinen, "A novel contextual topic model for multi-document summarization," *Expert Syst. Appl.*, vol. 42, no. 3, pp. 1340–1352, 2015, doi: 10.1016/j.eswa.2014.09.015.
- [4] H. Oliveira et al., "Assessing shallow sentence scoring techniques and combinations for single and multi-document summarization," *Expert Syst. Appl.*, vol. 65, pp. 68–86, 2016, doi: 10.1016/j.eswa.2016.08.030.
- [5] G. Glavaš and J. Šnajder, "Event graphs for information retrieval and multi-document summarization," *Expert Syst. Appl.*, vol. 41, no. 15, pp. 6904–6916, 2014, doi: 10.1016/j.eswa.2014.04.004.
- [6] Y. J. Kumar, N. Salim, A. Abuobieda, and A. T. Albaham, "Multi document summarization based on news components using fuzzy cross-document relations," *Appl. Soft Comput. J.*, vol. 21, pp. 265–279, 2014, doi: 10.1016/j.asoc.2014.03.041.
- [7] R. M. Alguliev, R. M. Aliguliyev, and M. S. Hajirahimova, "GenDocSum + MCLR: Generic document summarization based on maximum coverage and less redundancy," *Expert Syst. Appl.*, vol. 39, no. 16, pp. 12460–12473, 2012, doi: 10.1016/j.eswa.2012.04.067.
- [8] W. Luo, F. Zhuang, Q. He, and Z. Shi, "Exploiting relevance, coverage, and novelty for query-focused multi-document summarization," *Knowledge-Based Syst.*, vol. 46, pp. 33–42, 2013, doi: 10.1016/j.knosys.2013.02.015.
- [9] E. Barbu, M. T. Martín-Valdivia, E. Martínez-Cámara, and L. A. Ureña-López, "Language technologies applied to document simplification for helping autistic people," *Expert Syst. Appl.*, vol. 42, no. 12, pp. 5076–5086, 2015, doi: 10.1016/j.eswa.2015.02.044.
- [10] H. Oufaida, O. Nouali, and P. Blache, "Minimum redundancy and maximum relevance for single and multi-document Arabic text summarization," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 26, no. 4, pp. 450–461, 2014, doi: 10.1016/j.jksuci.2014.06.008.
- [11] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "Multiple documents summarization based on evolutionary optimization algorithm," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1675–1689, 2013, doi: 10.1016/j.eswa.2012.09.014.

- [12] R. M. Alguliev, R. M. Aliguliyev, and C. A. Mehdiyev, "Sentence selection for generic document summarization using an adaptive differential evolution algorithm," *Swarm Evol. Comput.*, vol. 1, no. 4, pp. 213–222, 2011, doi: 10.1016/j.swevo.2011.06.006.
- [13] R. M. Alguliev, R. M. Aliguliyev, and N. R. Isazade, "DESAMC+DocSum: Differential evolution with self-adaptive mutation and crossover parameters for multi-document summarization," *Knowledge-Based Syst.*, vol. 36, pp. 21–38, 2012, doi: 10.1016/j.knosys.2012.05.017.
- [14] J. U. Heu, I. Qasim, and D. H. Lee, "FoDoSu: Multi-document summarization exploiting semantic analysis based on social Folksonomy," *Inf. Process. Manag.*, vol. 51, no. 1, pp. 212–225, 2015, doi: 10.1016/j.ipm.2014.06.003.
- [15] J. P. Qiang, P. Chen, W. Ding, F. Xie, and X. Wu, "Multi-document summarization using closed patterns," *Knowledge-Based Syst.*, vol. 99, pp. 28–38, 2016, doi: 10.1016/j.knosys.2016.01.030.
- [16] K. Wu, L. Li, J. Li, and T. Li, "Ontology-enriched multi-document summarization in disaster management using submodular function," *Inf. Sci. (Ny)*, vol. 224, pp. 118–129, 2013, doi: 10.1016/j.ins.2012.10.019.
- [17] S. Xiong and D. Ji, "Query-focused multi-document summarization using hypergraph-based ranking," *Inf. Process. Manag.*, vol. 52, no. 4, pp. 670–681, 2016, doi: 10.1016/j.ipm.2015.12.012.
- [18] S. H. Zhong, Y. Liu, B. Li, and J. Long, "Query-oriented unsupervised multi-document summarization via deep learning model," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 8146–8155, 2015, doi: 10.1016/j.eswa.2015.05.034.
- [19] Y. Sankarasubramaniam, K. Ramanathan, and S. Ghosh, "Text summarization using Wikipedia," *Inf. Process. Manag.*, vol. 50, no. 3, pp. 443–461, 2014, doi: 10.1016/j.ipm.2014.02.001.
- [20] E. Canhasi and I. Kononenko, "Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization," *Expert Syst. Appl.*, vol. 41, no. 2, pp. 535–543, 2014, doi: 10.1016/j.eswa.2013.07.079.
- [21] D. R. Radev, H. Jing, M. Styś, and D. Tam, "Centroid-based summarization of multiple documents," *Inf. Process. Manag.*, vol. 40, no. 6, pp. 919–938, 2004, doi: 10.1016/j.ipm.2003.10.006.
- [22] D. Wang, S. Zhu, T. Li, Y. Chi, and Y. Gong, "Integrating Document clustering and multidocument summarization," *ACM Trans. Knowl. Discov. Data*, vol. 5, no. 3, 2011, doi: 10.1145/1993077.1993078.
- [23] X. Xu, "A New Sub-topics Clustering Method Based on Semi-supervised Learning," *JCP*, vol. 7, no. 10, pp. 2471–2478, 2012.

- [24] R. M. Aliguliyev, "Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization," *Comput. Intell.*, vol. 26, no. 4, pp. 420–448, 2010, doi: 10.1111/j.1467-8640.2010.00365.x.
- [25] Y. Xia, Y. Zhang, and J. Yao, "Co-clustering sentences and terms for multi-document summarization," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 6609 LNCS, no. PART 2, pp. 339–352, 2011, doi: 10.1007/978-3-642-19437-5_28.
- [26] Y. Nie, D. Ji, L. Yang, Z. Niu, and T. He, "Multi-document summarization using a clustering-based hybrid strategy," *Lect. notes Comput. Sci.*, vol. 4182, p. 608, 2006.
- [27] Z. Yang, K. Cai, J. Tang, L. Zhang, Z. Su, and J. Li, "Social context summarization," *SIGIR'11 - Proc. 34th Int. ACM SIGIR Conf. Res. Dev. Inf. Retr.*, pp. 255–264, 2011, doi: 10.1145/2009916.2009954.
- [28] J. Zhu et al., "Tag-oriented document summarization," *WWW'09 - Proc. 18th Int. World Wide Web Conf.*, no. 2007, pp. 1195–1196, 2009, doi: 10.1145/1526709.1526925.
- [29] S. Brin and L. Page, "Reprint of: The anatomy of a large-scale hypertextual web search engine," *Comput. Networks*, vol. 56, no. 18, pp. 3825–3833, 2012, doi: 10.1016/j.comnet.2012.10.007.
- [30] G. Erkan and D. R. Radev, "LexRank: Graph-based lexical centrality as salience in text summarization," *J. Artif. Intell. Res.*, vol. 22, pp. 457–479, 2004, doi: 10.1613/jair.1523.
- [31] S. Hariharan, T. Ramkumar, and R. Srinivasan, "Enhanced graph based approach for multi document summarization," *Int. Arab J. Inf. Technol.*, vol. 10, no. 4, pp. 334–341, 2013.
- [32] S. Hariharan and R. Srinivasan, "Studies on Graph Based Approaches for Single and Multi Document Summarizations," *Int. J. Comput. Theory Eng.*, vol. 1, no. 5, pp. 519–526, 2009, doi: 10.7763/ijcte.2009.v1.84.
- [33] X. Wan, "An exploration of document impact on graph-based multi-document summarization," *EMNLP 2008 - 2008 Conf. Empir. Methods Nat. Lang. Process. Proc. Conf. A Meet. SIGDAT, a Spec. Interes. Gr. ACL*, no. October, pp. 755–762, 2008, doi: 10.3115/1613715.1613811.
- [34] F. Wei, W. Li, Q. Lu, and Y. He, "A document-sensitive graph model for multi-document summarization," *Knowl. Inf. Syst.*, vol. 22, no. 2, pp. 245–259, 2010, doi: 10.1007/s10115-009-0194-2.
- [35] Y. Gong and X. Liu, "Generic text summarization using relevance measure and latent semantic analysis," in *Proceedings of the 24th annual international ACM SIGIR conference on research and development in information retrieval*, 2001, pp. 19–25.

- [36] J. H. Lee, S. Park, C. M. Ahn, and D. Kim, "Automatic generic document summarization based on non-negative matrix factorization," *Inf. Process. Manag.*, vol. 45, no. 1, pp. 20–34, 2009, doi: 10.1016/j.ipm.2008.06.002.
- [37] D. Wang, T. Li, and C. Ding, "Weighted feature subset non-negative matrix factorization and its applications to document understanding," in *Data Mining (ICDM)*, 2010 IEEE 10th International Conference on, 2010, pp. 541–550.
- [38] E. Baralis, L. Cagliero, S. Jabeen, A. Fiori, and S. Shah, "Multi-document summarization based on the Yago ontology," *Expert Syst. Appl.*, vol. 40, no. 17, pp. 6976–6984, 2013, doi: 10.1016/j.eswa.2013.06.047.
- [39] L. Hennig, W. Umbrath, and R. Wetzker, "An ontology-based approach to text summarization," *Proc. - 2008 IEEE/WIC/ACM Int. Conf. Web Intell. Intell. Agent Technol. - Work. WI-IAT Work.* 2008, pp. 291–294, 2008, doi: 10.1109/WIIAT.2008.175.
- [40] D. Wang and T. Li, "Weighted consensus multi-document summarization," *Inf. Process. Manag.*, vol. 48, no. 3, pp. 513–523, 2012, doi: 10.1016/j.ipm.2011.07.003.
- [41] J. Atkinson and R. Munoz, "Rhetorics-based multi-document summarization," *Expert Syst. Appl.*, vol. 40, no. 11, pp. 4346–4352, 2013, doi: 10.1016/j.eswa.2013.01.017.
- [42] G. Durrett, T. Berg-Kirkpatrick, and D. Klein, "Learning-based single-document summarization with compression and anaphoricity constraints," 54th Annu. Meet. Assoc. Comput. Linguist. ACL 2016 - Long Pap., vol. 4, pp. 1998–2008, 2016, doi: 10.18653/v1/p16-1188.
- [43] D. Bollegala, N. Okazaki, and M. Ishizuka, "A preference learning approach to sentence ordering for multi-document summarization," *Inf. Sci. (Ny)*, vol. 217, pp. 78–95, 2012, doi: 10.1016/j.ins.2012.06.015.
- [44] D. Bollegala, N. Okazaki, and M. Ishizuka, "A bottom-up approach to sentence ordering for multi-document summarization," *Inf. Process. Manag.*, vol. 46, no. 1, pp. 89–109, 2010, doi: 10.1016/j.ipm.2009.07.004.
- [45] R. Ferreira, R. D. Lins, S. J. Simske, F. Freitas, and M. Riss, "Assessing sentence similarity through lexical, syntactic and semantic analysis," *Comput. Speech Lang.*, vol. 39, pp. 1–28, 2016, doi: 10.1016/j.csl.2016.01.003.
- [46] E. ShafieiBavani, M. Ebrahimi, R. Wong, and F. Chen, "On Improving Informativity and Grammaticality for Multi-Sentence Compression," no. November, 2016, [Online]. Available: <http://arxiv.org/abs/1605.02150>.
- [47] L. Marujo et al., "Exploring events and distributed representations of text in multi-document summarization," *Knowledge-Based Syst.*, vol. 94, pp. 33–42, 2016, doi: 10.1016/j.knosys.2015.11.005.

- [48] E. Baralis, L. Cagliero, N. Mahoto, and A. Fiori, "GraphSum: Discovering correlations among multiple terms for graph-based summarization," *Inf. Sci. (Ny)*, vol. 249, pp. 96–109, 2013, doi: 10.1016/j.ins.2013.06.046.
- [49] Y. Chali, S. A. Hasan, and S. R. Joty, "Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels," *Inf. Process. Manag.*, vol. 47, no. 6, pp. 843–855, 2011, doi: 10.1016/j.ipm.2010.10.002.
- [50] C. Sunitha, A. Jaya, and A. Ganesh, "A Study on Abstractive Summarization Techniques in Indian Languages," *Procedia Comput. Sci.*, vol. 87, pp. 25–31, 2016, doi: 10.1016/j.procs.2016.05.121.
- [51] A. John and M. Wilsy, "Vertex cover algorithm based multi-document summarization using information content of sentences," *Procedia Comput. Sci.*, vol. 46, no. Icict 2014, pp. 285–291, 2015, doi: 10.1016/j.procs.2015.02.022.
- [52] E. Canhasi and I. Kononenko, "Weighted hierarchical archetypal analysis for multi-document summarization," *Comput. Speech Lang.*, vol. 37, pp. 24–46, 2016, doi: 10.1016/j.csl.2015.11.004.
- [53] Y. Zhang, Y. Xia, Y. Liu, and W. Wang, "Clustering sentences with density peaks for multi-document summarization," *NAACL HLT 2015 - 2015 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Conf.*, pp. 1262–1267, 2015, doi: 10.3115/v1/n15-1136.
- [54] E. Tzouridis, J. Nasir, and U. Brefeld, "Learning to summarise related sentences," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 1636–1647.
- [55] N. K. Nagwani, "Summarizing large text collection using topic modeling and clustering based on MapReduce framework," *J. Big Data*, vol. 2, no. 1, pp. 1–18, 2015, doi: 10.1186/s40537-015-0020-5.
- [56] P. Sukumar and K. S. Gayathri, "Semantic based Sentence Ordering Approach for Multi-Document Summarization," *Int. J. Recent Technol. Eng.*, no. 2, pp. 2277–3878, 2014.
- [57] J. Nasir, A. Karim, G. Tsatsaronis, and I. Varlamis, "A knowledge-based semantic kernel for text classification," in *String Processing and Information Retrieval*, 2011, pp. 261–266.
- [58] J. Carbonell and J. Goldstein, "Use of MMR, diversity-based reranking for reordering documents and producing summaries," *SIGIR Forum (ACM Spec. Interes. Gr. Inf. Retrieval)*, pp. 335–336, 1998, doi: 10.1145/3130348.3130369.
- [59] J. Lin, N. Madnani, and B. J. Dorr, "Putting the user in the loop: interactive maximal marginal relevance for query-focused summarization," in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, 2010, pp. 305–308.

- [60] J. Christensen, S. Soderland, G. Bansal, and Mausam, "Hierarchical summarization: Scaling up multi-document summarization," 52nd Annu. Meet. Assoc. Comput. Linguist. ACL 2014 - Proc. Conf., vol. 1, no. Figure 1, pp. 902–912, 2014, doi: 10.3115/v1/p14-1085.
- [61] D. Wang, S. Zhu, T. Li, and Y. Gong, "Multi-document summarization using sentence-based topic models," ACL-IJCNLP 2009 - Jt. Conf. 47th Annu. Meet. Assoc. Comput. Linguist. 4th Int. Jt. Conf. Nat. Lang. Process. AFNLP, Proc. Conf., no. August, pp. 297–300, 2009, doi: 10.3115/1667583.1667675.
- [62] A. Rodriguez and A. Laio, "Clustering by fast search and find of density peaks," *Science* (80-.), vol. 344, no. 6191, pp. 1492–1496, 2014, doi: 10.1126/science.1242072.
- [63] B. Wang, J. Zhang, Y. Liu, and Y. Zou, "Density peaks clustering based integrate framework for multi-document summarization," *CAAI Trans. Intell. Technol.*, vol. 2, no. 1, pp. 26–30, 2017, doi: 10.1016/j.trit.2016.12.005.
- [64] M. G. Ozsoy, I. Cicekli, and F. N. Alpaslan, "Text summarization of turkish texts using latent semantic analysis," in *Proceedings of the 23rd international conference on computational linguistics*, 2010, pp. 869–876.
- [65] J. Christensen, Mausam, S. Soderland, and O. Etzioni, "Towards coherent multi-document summarization," *NAACL HLT 2013 - 2013 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Main Conf.*, no. June, pp. 1163–1173, 2013.
- [66] D. Contractor, Y. Guo, and A. Korhonen, "Using argumentative zones for extractive summarization of scientific articles," 24th Int. Conf. Comput. Linguist. - Proc. COLING 2012 Tech. Pap., no. December, pp. 663–678, 2012.
- [67] N. Chatterjee and N. Yadav, "Fuzzy Rough Set-Based Sentence Similarity Measure and its Application to Text Summarization," *IETE Tech. Rev. (Institution Electron. Telecommun. Eng. India)*, vol. 36, no. 5, pp. 517–525, 2019, doi: 10.1080/02564602.2018.1516521.
- [68] J. Xu and G. Durrett, "Neural extractive text summarization with syntactic compression," *EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf.*, pp. 3292–3303, 2020, doi: 10.18653/v1/d19-1324.
- [69] E. Canhasi, "Query Focused Multi-document Summarization Based on Five-Layered Graph and Universal Paraphrastic Embeddings," in *Computer Science On-line Conference*, 2017, pp. 220–228.
- [70] R. Rautray and R. C. Balabantaray, "An evolutionary framework for multi document summarization using Cuckoo search approach: MDSCSA," *Appl. Comput. Informatics*, vol. 14, no. 2, pp. 134–144, 2018, doi: 10.1016/j.aci.2017.05.003.
- [71] C. Ding, X. He, and H. D. Simon, "On the equivalence of non-negative matrix factorization and spectral clustering," *Proc. 2005 SIAM Int. Conf. Data Mining, SDM 2005*, no. 4, pp. 606–610, 2005, doi: 10.1137/1.9781611972757.70.

- [72] Pattanaik, A., Sagnika, S., Das, M., & Mishra, B. S. P. (2019). Extractive summary: an optimization approach using bat algorithm. In *Ambient communications and computer systems* (pp. 175-186). Springer, Singapore. doi: 10.1007/978-981-13-5934-7.
- [73] P. Verma and H. Om, "MCRMR: Maximum coverage and relevancy with minimal redundancy based multi-document summarization," *Expert Syst. Appl.*, vol. 120, pp. 43–56, 2019, doi: 10.1016/j.eswa.2018.11.022.
- [74] A. Naserasadi, H. Khosravi, and F. Sadeghi, "Extractive multi-document summarization based on textual entailment and sentence compression via knapsack problem," *Nat. Lang. Eng.*, vol. 25, no. 1, pp. 121–146, 2019, doi: 10.1017/S1351324918000414.
- [75] A. Joshi, E. Fidalgo, E. Alegre, and L. Fernández-Robles, "SummCoder: An unsupervised framework for extractive text summarization based on deep auto-encoders," *Expert Syst. Appl.*, vol. 129, pp. 200–215, 2019, doi: 10.1016/j.eswa.2019.03.045.
- [76] Mahajani, A., Pandya, V., Maria, I., & Sharma, D. (2019). A comprehensive survey on extractive and abstractive techniques for text summarization. *Ambient communications and computer systems*, 339-351.
- [77] Gupta, V., Bansal, N., & Sharma, A. (2019). Text summarization for big data: A comprehensive survey. In *International Conference on Innovative Computing and Communications* (pp. 503-516). Springer, Singapore.
- [78] A. Kanapala, S. Pal, and R. Pamula, "Text summarization from legal documents: a survey," *Artif. Intell. Rev.*, vol. 51, no. 3, pp. 371–402, 2019, doi: 10.1007/s10462-017-9566-2.
- [79] A. Kumar and A. Sharma, "Systematic literature review of fuzzy logic based text summarization," *Iran. J. Fuzzy Syst.*, vol. 16, no. 5, pp. 45–59, 2019, doi: 10.22111/ijfs.2019.4906.
- [80] G. Erkan and D. R. Radev, "LexPageRank: Prestige in Multi-Document Text Summarization," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, 2004.
- [81] Jalil, Z., J.A. Nasir, and M. Nasir, *Extractive Multi-Document Summarization: A Review of Progress in the Last Decade*. IEEE Access, 2021.
- [82] Filippova, K. (2010, August). Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 322-330). Association for Computational Linguistics.
- [83] Wang, L., & Ling, W. (2016). Neural Network-Based Abstract Generation for Opinions and Arguments. *arXiv preprint arXiv:1606.02785*.
- [84] Nenkova, A. (2005, July). Automatic text summarization of newswire: Lessons learned from the document understanding conference. In *AAAI* (Vol. 5, pp. 1436-1441).

- [85] Bidoki, M., M.R. Moosavi, and M. Fakhrahmad, A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities. *Information Processing & Management*, 2020. 57(6): p. 102341.
- [86] Sanchez-Gomez, J.M., M.A. Vega-Rodríguez, and C.J. Perez, A decomposition-based multi-objective optimization approach for extractive multi-document text summarization. *Applied Soft Computing*, 2020. 91: p. 106231.
- [87] El-Kassas, W.S., et al., Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 2021. 165: p. 113679.
- [88] Mojrian, M. and S.A. Mirroshandel, A novel extractive multi-document text summarization system using quantum-inspired genetic algorithm: MTSQIGA. *Expert Systems with Applications*, 2021. 171: p. 114555.
- [89] Siautama, R., A.C. IA, and D. Suhartono, Extractive hotel review summarization based on TF/IDF and adjective-noun pairing by considering annual sentiment trends. *Procedia Computer Science*, 2021. 179: p. 558-565.
- [90] Uçkan, T. and A. Karci, Extractive multi-document text summarization based on graph independent sets. *Egyptian Informatics Journal*, 2020. 21(3): p. 145-157.
- [91] Chen, J. and H. Zhuge, Extractive summarization of documents with images based on multi-modal RNN. *Future Generation Computer Systems*, 2019. 99: p. 186-196.
- [92] Celikkale, B., et al., Generating visual story graphs with application to photo album summarization. *Signal Processing: Image Communication*, 2021. 90: p. 116033.
- [93] Shingrakhia, H. and H. Patel, SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization. *The Visual Computer*, 2021: p. 1-17.
- [94] Radarapu, R., A.S.S. Gopal, and N. Madhusudhan, Video summarization and captioning using dynamic mode decomposition for surveillance. *International Journal of Information Technology*, 2021: p. 1-10.
- [95] Marzijarani, S.B. and H. Sajedi, Opinion mining with reviews summarization based on clustering. *International Journal of Information Technology*, 2020. 12(4): p. 1299-1310.
- [96] Abdi, A., et al., A hybrid deep learning architecture for opinion-oriented multi-document summarization based on multi-feature fusion. *Knowledge-Based Systems*, 2021. 213: p. 106658.
- [97] Pontes, E.L., et al., Compressive approaches for cross-language multi-document summarization. *Data & Knowledge Engineering*, 2020. 125: p. 101763.
- [98] El-Kassas, W.S., et al., EdgeSumm: Graph-based framework for automatic text summarization. *Information Processing & Management*, 2020. 57(6): p. 102264.
- [99] Wang, D., et al., Heterogeneous graph neural networks for extractive document summarization. *arXiv preprint arXiv:2004.12393*, 2020.
- [100] Tomer, M. and M. Kumar, Multi-document extractive text summarization based on firefly algorithm. *Journal of King Saud University-Computer and Information Sciences*, 2021.

- [101] Davoodijam, E., et al., MultiGBS: A multi-layer graph approach to biomedical summarization. *Journal of Biomedical Informatics*, 2021. 116: p. 103706.
- [102] Jin, H., T. Wang, and X. Wan. Multi-granularity interaction network for extractive and abstractive multi-document summarization. in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 2020.
- [103] Van Lierde, H. and T.W. Chow, Query-oriented text summarization based on hypergraph transversals. *Information Processing & Management*, 2019. 56(4): p. 1317-1338.
- [104] Li, X., et al., Sentiment Lossless Summarization. *Knowledge-Based Systems*, 2021: p. 107170.
- [105] Mallick, C., et al., Graph-based text summarization using modified TextRank, in *Soft computing in data analytics*. 2019, Springer. p. 137-146.
- [106] Blei, D.M., A.Y. Ng, and M.I. Jordan, Latent dirichlet allocation. *the Journal of machine Learning research*, 2003. 3: p. 993-1022.
- [107] Teh, Y.W., et al., Hierarchical dirichlet processes. *Journal of the american statistical association*, 2006. 101(476): p. 1566-1581.
- [108] Swapna, B. and T. Anuradha. Achieving Higher Ranking to Webpages Through Search Engine Optimization. in *Proceedings of International Conference on Computational Intelligence and Data Engineering*. 2018. Springer.
- [109] Hirao, T., et al. Extracting important sentences with support vector machines. in *COLING 2002: The 19th International Conference on Computational Linguistics*. 2002.
- [110] Saura, J.R., Using data sciences in digital marketing: Framework, methods, and performance metrics. *Journal of Innovation & Knowledge*, 2021. 6(2): p. 92-102.
- [111] Kianmehr, K., et al. Text summarization techniques: SVM versus neural networks. in *Proceedings of the 11th International Conference on Information Integration and Web-based Applications & Services*. 2009.
- [112] Mihalcea, R. and P. Tarau. Textrank: Bringing order into text. in *Proceedings of the 2004 conference on empirical methods in natural language processing*. 2004.
- [113] Mabroukeh, N. R., & Ezeife, C. I. (2010). A taxonomy of sequential pattern mining algorithms. *ACM Computing Surveys (CSUR)*, 43(1), 1-41.
- [114] Do, T. D. T., Laurent, A., & Termier, A. (2010, December). Pglcm: Efficient parallel mining of closed frequent gradual itemsets. In *2010 IEEE International Conference on Data Mining* (pp. 138-147). IEEE.
- [115] Song, Y., Ng, W., Leung, K. W. T., & Fang, Q. (2015). SFP-Rank: significant frequent pattern analysis for effective ranking. *Knowledge and Information Systems*, 43(3), 529-553.

- [116] Baralis, E., Cagliero, L., Jabeen, S., & Fiori, A. (2012, March). Multi-document summarization exploiting frequent itemsets. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing* (pp. 782-786).
- [117] Carbonell, J., & Goldstein, J. (1998, August). The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 335-336).
- [118] Marcu, D. (1999). Discourse trees are good indicators of importance in text. *Advances in automatic text summarization*, 293, 123-136.
- [119] Chengcheng, L. (2010, October). Automatic text summarization based on rhetorical structure theory. In *2010 International Conference on Computer Application and System Modeling (ICCSM 2010)* (Vol. 13, pp. V13-595). IEEE.
- [120] Ibrahim, A., Elghazaly, T., & Gheith, M. (2013). A novel Arabic text summarization model based on rhetorical structure theory and vector space model. *International Journal of Computational Linguistics and Natural Language Processing*, 2(8), 480-485.
- [121] <https://www.slideshare.net/LiaRatna1/sinonim-38250183> Web link 2019
- [122] Sinha, A., Yadav, A., & Gahlot, A. (2018). Extractive text summarization using neural networks. *arXiv preprint arXiv:1802.10137*.
- [123] Nasir, M., Ikram, N., & Jalil, Z. (2022). Usability inspection: Novice crowd inspectors versus expert. *Journal of Systems and Software*, 183, 111122.
- [124] Jalil, Z., Nasir, M., Alazab, M., Nasir, J., Amjad, T., & Alqammaz, A. (2023). Grapharizer: A Graph-Based Technique for Extractive Multi-Document Summarization. *Electronics*, 12(8), 1895.

Appendix B

Sample Summaries

Sample Summaries

The Reference summaries and different system summaries are given below.

Grapharizer

“Cambodian leader Sam_Rainsy a staunch critic of Hun_Sen was forced to take refuge in a UN office in September to avoid arrest after Hun_Sen accused Sam_Rainsy of being behind a plot against Hun_Sen life. Sok_An representing Hun_Sens party said Friday that one working group had completed work on a joint political platform to be implemented by the new government. The remaining senators Sihanouk said should be selected by a method agreed upon by the new government and the NA. Hun_Sen announced a government guarantee Wednesday of all politicians safety and right to conduct political activities in accordance with the laws. I dont think there is any benefit for Hun_Sen to cause instability for our country Ranariddh said. Hun_Sens got the premiership and legitimacy through the election and recognition from Sihanouk. The deal which will make Hun_Sen prime minister and Ranariddh president of the NA ended more than three months of political deadlock that followed a July election narrowly won by Hun_Sen. Sihanouk recalling procedures used in a past government suggested Tuesday that Sihanouk should appoint the first two members of the upper house. Ranariddhs ally Sam_Rainsy whose party placed a distant third in the election was excluded of last weeks deal. Senior FUNCINPEC official Ahmad_Yahya revealed Monday that Senior FUNCINPEC official Ahmad_Yahya was also agreed that CPP would control the foreign affairs and finance portfolios”.

SVMGrapharizer

“Fearing arrest many opposition_leaders of Parliament left Cambodia after the ceremonial opening of the NA on Sep_24. Co-sharing anything with the Cambodian Peoples Party means surrendering full power to the Cambodian Peoples Party. Hun_Sen said Monday that the CPP and FUNCINPEC had agreed that the Senate would be half as large as the 122-seat NA. Hun_Sen has guaranteed the safety and political freedom of all politicians trying to ease the fears of Hun_Sen rivals that they will be arrested or killed if they return to the country. But The Sam_Rainsy Party refuses to negotiate. Sam_Rainsy who earlier called Hun_Sens statement full of loopholes asked Sihanouk for help in obtaining a promise from Hun_Sen that all members of the Sam_Rainsy Party were free from prosecution for all members of the Sam_Rainsy Party. Political activities during and after last Julys election the opposition alleging widespread fraud and intimidation refused to accept the results of the polls. In a short letter sent to news agencies Sihanouk said Sihanouk had received copies of cooperation agreements signed on Monday that will place Hun_Sen and Sihanouk Cambodian Peoples Party in firm control of fiscal and administrative functions in the government. Disputes over the presidency of Parliament have been a major Hurdle in talks between the

opposition block and the Cambodian Peoples Party to form a new government. Hun_Sen and Ranariddh in a coalition formed in 1993 after a landmark UN-sponsored election often clashed over power-sharing and the integration of guerrilla fighters from the crumbling Khmer_Rouge”.

MLRGrapharizer

“The united nations offices in a vocally anti cambodian leader hun sen opposition parties leaders of hun_sen was forced to kingdom of cambodia and people king norodom sihanouk wrote. Sok An representing hun sen's party in a joint political platform to the new government. Cambodian leader hun sen and the new government at a method agreed nov. Fearing for safety and political freedom of the laws in a statement. Think cambodian leader hun sen to threaten the former finance minister sam ram rainsy party negotiating partners with arrest for country prince norodom ranariddh said. Former finance minister sam ram rainsy and loopholes asked king norodom sihanouk for all members of the upper house of parliament. The upper house of parliament to be led by cambodian leader hun sen and chief rival prince norodom ranariddh president of political deadlock that followed a new government return. King norodom sihanouk recalling procedures used in a coalition government led by the two political parties. The opposition parties bloc and struck a deal with cambodian leader hun sen to form a new government. Cambodian leader hun sen and prince norodom ranariddh ranariddhs in a copy of kingdom of Cambodia”.

ANNGrapharizer

“leader Sam_Rainsya staunch critic of Hun_Sen was forced to take refuge in a UN office in September to avoid arrest after Hun_Sen accused him of being behind a plot against his life. Sok_An representing Hun_Sen's party said Friday that one worSihanouk group had completed one worSihanouk group work on a joint political platform to be implemented by the new government. The remaining senators Sihanouk said should be selected by a method agreed upon by the new government and the NA. Hun_Sen announced a government guarantee Wednesday of all politicians safety and all politicians right to conduct political activities in accordance with the laws in force. Worried that party colleagues still face arrest for party colleagues politics opposition leader Sam_Rainsy sought further clarification Friday of security guarantees promised by Hun_Sen. The two parties have formed three worSihanouk groups to settle details of the agreement including the establishment of a Senate to be the Senate. the opposition alleging widespread fraud and intimidation refused to accept the results of the polls. Sihanouk recalling procedures used in a past government suggested Tuesday that Sihanouk should appoint the first two members of the upper house. No party with seats at the NA should be excluded if transparency and inspection system is to be used in government the faxed statement said adding that the Cambodian Peoples

Party should also be given chairmanship of one of the NAs nine commissions. Khmer Rouge was responsible for the deaths of as many as 2 million people during the guerrilla groups 1975-79 rule of Cambodia”.

Term Based

“The killings of nearly 100 of Ranariddh's supporters documented by U.N. human rights workers in the aftermath of the coup were dismissed by the CPP as mostly fabrications meant to distort the political situation. The Khmer Rouge was responsible for the deaths of as many as 2 million people during the guerrilla group's 1975-79 rule of Cambodia. The monitoring ended Sept. 30. The ruling party also reminded the United States that Washington supported a Cambodian exile government dominated by the brutal Khmer Rouge in the 1980s. At least four demonstrators were killed by police, but the discovery of more than 20 bodies in the aftermath has prompted speculation that the death tally could be much higher. U.N. human rights workers later discovered more than 20 bodies _ many bearing signs of torture _ in and around the capital, prompting speculation that the death toll could be much higher. In the most recent elections, held in July, Hun Sen's party collected 64 of the 122 parliamentary seats, but was short of the two-thirds majority needed to set up a new government. Hun Sen's Cambodian People's Party won 64 of the 122 parliamentary seats in July's elections, short of the two-thirds majority needed to form a government on its own. Hun Sen's party won 64 of the 122 seats in parliament in July's national election, but not the two-thirds majority necessary to form a government on its own”.

YAGO Ontology

“Hun Sen, however, rejected that. Hun Sen said on Friday that the opposition concerns over their safety in the country was “just an excuse for them to stay abroad.” Hun Sen was not home at the time of the attack, which was followed by a police crackdown on demonstrators contesting Hun Sen's election victory. Hun Sen said Monday that the CPP and FUNCINPEC had agreed that the Senate would be half as large as the 122-seat National Assembly. Hun Sen used Thursday's anniversary of a peace agreement ending the country's civil war to pressure the opposition to form a coalition government with his party. “Only those who want to prolong the anarchy and instability prevent efforts to set up a new government,” Hun Sen said in a televised speech marking the anniversary of the 1991 Paris Peace Accords. Hun Sen said his current government would remain in power as long as the opposition refused to form a new one. Hun Sen has rejected the opposition's reservations, saying it would be inappropriate to hold a summit outside the country. Hun Sen implied Thursday that the opposition failed to follow through on promises made at the summit.

“If those results are strictly respected, there seems no reason to hold another summit,” Hun Sen said in a speech on the anniversary of the 1991 Paris Peace Accords”.

Closed Patterns

“Hun Sen blamed the violence on opposition leaders, saying the demonstrations instigated social and economic chaos. Less than two weeks after abandoning hope that he could influence the parties to reach a compromise, Sihanouk is now “strongly interested” in presiding over a summit meeting of the three party leaders in Cambodia, Machimura said. Diplomatic efforts to revive the stalled talks appeared to bear fruit Monday as Japanese Foreign Affairs Secretary of State Nobutaka Machimura said King Norodom Sihanouk has called on Ranariddh and Sam Rainsy to return to Cambodia. “The leaders of illegal demonstrations are the ones who must bear responsibility for the consequences deriving from the protest,” the CPP said Tuesday, referring to the deadly violence as “minor incidents.” After a meeting between Hun Sen and the new French ambassador to Cambodia, Hun Sen aide Prak Sokhonn said the Cambodian leader had repeated calls for the opposition to return, but expressed concern that the international community may be asked for security guarantees. “Our office has not received any official request for that operation to be started up again,” U.N. diplomat Jonathan Prentice said Monday in reaction to Prak Sokhonn’s statement. The vote failed to put an end to instability that followed last year’s coup. The king, the sole force in Cambodian politics able to broker a deal, pressured both sides to reach agreement before he leaves Saturday for medical treatment in Beijing. After a series of border clashes, the Khmer Rouge was ousted from power by an invading Vietnamese army that set up a surrogate Cambodian communist government later led by Hun Sen. Hun Sen’s ruling party narrowly won a majority in elections in July, but the opposition claiming widespread intimidation and fraud has denied Hun Sen the two-thirds vote in parliament required to approve the next government”.