Summarization of Opinions from Multiple Documents

By

Rubab Hafeez

2018-NUST-MS-IT-00000119474

Supervisor

Dr. Sharifullah Khan

Department of Computing

A thesis submitted in partial fulfillment of the requirements for the degree of Masters of Science in Information Technology (MS IT)

In

School of Electrical Engineering and Computer Science,
National University of Sciences and Technology (NUST),
Islamabad, Pakistan.

(Nonember 2018)

Approval

It is certified that the contents and form of the thesis entitled "**Summarization of Opinions from Multiple Documents**" submitted by **Rubab Hafeez** have been found satisfactory for the requirement of the degree.

Advisor: **Dr. Sharifullah Khan**
Signature: __________________

Date: __________________

Committee Member 1: **Dr. Azeem Abbas Khan**

Signature: __________________
Date: __________________

Committee Member 2: **Dr. Safdar Abbas Khan**

Signature: __________________
Date: __________________

Committee Member 3: **Dr. Anee-ur-Rehman**

Signature: __________________
Date: __________________

Abstract

Daily amount of news reporting in real-world events is growing exponentially, at the same time, people need most important information about any event or any topic in an organized or compact form to make decisions. Document summarization addresses the problem of presenting the information in a compact form to the readers. Different approaches to summarize documents have been proposed and evaluated in literature. Common research problems in summarization are redundancy and extraction of sentences; that are important and semantically linked with other sentences.

The proposed summarization approach is a combination of agglomerative hierarchical clustering and Latent Semantic Analysis (LSA); which measures the semantic similarity among different terms and reduces dimensions by preserving only highly weighted vectors, we propose a novel multi document summarization approach. To identify important terms in our summary, we have used Latent Dirichlet Allocation Model (LDA). LDA is a generative statistical model which allows a set of observations to be explained by a set of small number of topics, where the presence of each word is attributable to the topics of the documents.

We have used Recall Oriented Understudy for Gisting Evaluation (ROUGE) metric for the evaluation of our system against other state-of-the art techniques using Document Understanding Conference (DUC) dataset 2004. Experimental results show that there is substantial performance improvement using our system and it makes a coherent summary as compared to the other state-of-art techniques. Our summarization approach improves upon current state-of-the-art summarization systems on mainstream evaluation datasets.

Dedication

This thesis is dedicated to my lovable parents Mr. Muhammad Hafeez Khan and Mrs. Shamim Hafeez who provided me support and confidence to face the difficulties of life and courage to fulfil my dreams.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgement has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation and linguistics which has been acknowledged.

Author Name: **Rubab Hafeez**
Signature: ________________

Acknowledgment

I pay my gratitude to Allah Almighty for blessing me a lot and without His guidance I was not able to complete this research. I would like to dedicate my thesis to my parents, especially to my father Mr. Muhammad Hafeez Khan, my mother and my siblings who supported me and made me see this day.

I desire, to express my sincere thanks to my Supervisor Dr. Sharifullah Khan, who being a mentor provided me a constant guidance and support that empowered me to finish this work within due course of time. His supervision assisted me in all the dimensions of this thesis work. I could not have anticipated having a better advisor for my MS research. He inspired and encouraged me throughout my journey in bringing this assigned task to an adequate completion. His timely and efficient contributions helped me to shape the thesis into its final form and I express my gratefulness for his sincere supervision all the way.

I am also obliged to Dr. Azeem Abbas, Dr. Safdar Abbas and Dr. Anees-ur-Rehman in SEECS NUST for the unremitting assistance, treasured guidance and recommendations in the technical and experimental work. I am extremely thankful to, Dean/Principal of SEECS Dr. Syed Hassan Zaidi for providing me with all the vital facilities and the commendable atmosphere for the research.

Table of Contents

1 Introduction and Motivation 1

1.1 Introduction 1

1.2 Motivation 4

1.3 Problem Statement 4

1.3.1 Repetition of letters 5

1.3.2 Increased feature space 5

1.3.3 Summary about a central topic 5

1.3.4 Contributing topic terms 5

1.4 Research Objectives 5

1.5 Thesis Outline 6

2 Literature Review 7

2.1 Single Document Summarization 8

2.2 Feature based approaches 8

2.3 Machine learning approaches 9

2.4 Multi-document Summarization 12

2.4.1 Clustering based Approaches 12

2.4.2 Summarization using Latent Semantic Analysis 13

2.4.3 Graph based Approaches 14

2.4.4 Topic Modeling Summarization 14

2.4.5 Critical Analysis 15

3 The Proposed Framework 19

3.1 Data Acquisition 19

3.1.1 Data Description 19
3.2 Proposed Framework 20
3.3 Data Loading 20
3.4 Data Pre-processing 20
3.4.1 Tokenization 21
3.4.2 Stop Words Removal 21
3.4.3 Punctuation Removal 22
3.5 Dimensions reduction using Latent Semantic Analysis (LSA) . 23
3.5.1 Vector Space Representation 24
3.5.2 Singular Value Decomposition (SVD) 25
3.6 Hierarchical Agglomerative Clustering (HAC) 26
3.6.1 Bottom-up approach 26
3.7 Summary generation 27
3.7.1 Sentence selection based upon tfidf scores 27
3.8 Topic Detection 28
3.8.1 Latent dirichlet Allocation Model (LDA) 28
3.9 Algorithm for Proposed framework 29

4 Experiments and Evaluation 31
4.1 Evaluation Metrics 31
4.2 Data Specification 32
4.3 Experiment 1 33
4.3.1 Sample of data set 33
4.3.2 Preprocessing results 34
4.3.3 Result sample for tokeniztaion 34
4.3.4 Result sample for stopwords and punctuation removal . 35
4.3.5 Result sample of Vector space modeling 36
4.3.6 Dimension Reduction Results 36
4.3.7 Result sample of $tfidf$ 36
4.3.8 Result sample of LSA 37
4.3.9 Results of clustering 37
4.3.10 Result sample of HAC 37
4.3.11 Summary generation 38

4.3.12 Result sample of summary:1 39
4.3.13 Result sample of summary:2 40
4.3.14 Topic detection results 40
4.3.15 Result sample for topic detection; summary1 41
4.3.16 Result sample for topic detection; summary2 41
4.3.17 Evaluation Results on Experiment-1 41
4.3.18 Results on: ROUGE-1 42
4.3.19 Results on: ROUGE-2 43
4.3.20 Results on: ROUGE-SU4 44
4.4 Experiment 2 45
4.4.1 Results on module 1: preprocessing 46
4.4.2 Punctuation Results on Experiment:2 46
4.4.3 Vector Space Representation results on Experiment:2 . 46
4.4.4 Results on module: Dimension Reduction 47
4.4.5 Latent Semantic Analysis results on Experiment:2 . . . 47
4.4.6 Clustering results on Experiment:2 48
4.4.7 Summary generation on Experiment:2 48
4.4.8 Topic Detection on Experiment:2 49
4.4.9 Evaluation Results on Experiment-2 49
4.4.10 Evaluation results on Experiment:2 49

5 Conclusion and Future work 51
5.1 Conclusion 51
5.1.1 Contribution of the Research 51
5.1.2 Discussion 52
5.1.3 Future Work 53

List of Figures

2.1 Cosine Metric for computing similarity to title [9] 10

3.1 Proposed System Model for summary generation 21

4.1 Evaluation scores on Rouge-1: dataset(DUC'04) 42
4.2 Evaluation scores on Rouge-2: dataset(DUC'04) 43
4.3 Evaluation scores on Rouge-SU4: dataset(DUC'04) 44

List of Tables

1.1 Size of the web over 3 years 3

2.1 Different types of summarization systems 7
2.2 Machine learning trainable features 9
2.3 Machine learning approaches 11
2.4 Comparision of previous research approaches 17

3.1 Specification of TASKS in DUC 2004 data set 20
3.2 Effects of preprocessing on Bag of Words feature space 23

4.1 Document term matrix, result sample 36
4.2 Document term matrix, result sample 36
4.3 Specification of the Benchmark techniques 41
4.4 Results on DUC Peer code (A-F) 45
4.5 Results on DUC Peer code (W-Z) 45
4.6 Data set for experiment:2 45
4.7 Tokenization results for experiment:2 46
4.8 Punctuation results for experiment:2 46
4.9 Document term matrix for experiment:2 47
4.10 tfidf matrix for experiment:2 47
4.11 LSA for experiment:2 48
4.12 Cluster for experiment:2 48
4.13 Summary topics for experiment:2 49
4.14 Specification of the Benchmark techniques 49
4.15 ROUGE Scores 50

5.1 Contribution of the proposed framework 52

Chapter 1

Introduction and Motivation

With the rapid growth in the web size, it has become need of the hour to develop information retrieval system which provides an easy and effecient access to the information for all the users. Summarization is one of many such information retrieval systems that can address the problem of information overload. Automatic summarization system generates a summary of related textual documents providing an overall understanding of the topic, without having to go through all of the documents [63].
This chapter further explains the need of such a summarization system, motivation behind this research in section 1.2. Section 1.3 contains problem statement and Section 1.4 describes objectives of this research. Section 1.5 provides an overview of all the chapters with contents.

1.1 Introduction

Summarization is a process of creating a topic-focused or generic summary by reducing the size of a document while keeping the main and central characteristics of documents intact. Summarization helps the users in saving their time and to give them much information in a reduced form. The estimated size of the web is 4.48 billion pages till October 2018 [46]. This number grows at a very fast pace in terms of articles, books and scientific pages every day, such that it is unfeasible to sieve effective and useful information which arouse the need to summarize documents into a compressed and precise form. Summarization helps a reader to decide whether a document is related to his/her interest or not [60].

The field of summarization has been investigated by Natural Language Processing (NLP) community for nearly half a century and since 1958; it has been producing essential information from a large collection of data sets [59]. Primarily research focused on a summarization of single documents, where summary is created from a single document. A document revolves around a central topic or theme. Summary arranges sentences in such a way that irrelevant or redundant data is removed and the summary contains information about a main topic from which a reader can get information about the gist of the document, he doesn't need to read the original document entirely. His time is saved as he can get the primary information in lesser time span. In single document summarization, the sentences are selected to form a summary based on any weighting scheme, i.e, term frequency ($tfidf$) defined in the following equation.

$$IDF(t) = log\_e(Totalnumberofdocuments/Numberofdocumentswithtermtinit). \tag{1.1}$$

where similarity between two sentences is measured by a function of content overlap. Overlap can be seen as a number of similar tokens between the lexical representations of sentences [7]. These summarization algorithms are about measuring word frequencies or some other combinations of measuring weights and generating summary by combining sentences which have higher frequencies. These weighting mechanisms have limited choice in terms of assigning weights to the sentences or terms [3].

Graph based approaches are also used in summarizing both single and multiple documents. In graph based approaches, sentences are represented as nodes in a directed or an undirected graph. Two sentences are connected together with edges only if they share some similarity (cosine or such) behind some predefined threshold [24]. Machine learning approaches are supervised techniques in which classifier, i.e, Neural Networks, etc are trained on certain features. Sentences owing to have such features are extracted to form a summary [24].

The web size is growing at a large pace, the size of the web in previous three years is shown in Table 1.1.

Table 1.1: Size of the web over 3 years

Time-frame	**Size**
2018	4.45 billion pages
2017	3.39 billion pages
2016	2.69 billion pages

With this continuous growth in web data, it became a necessity to process multiple documents. In multi-document summarization (MDS), summary is created across multiple documents by reducing them in size while preserving the central characteristics of original documents. The major challenge with MDS is that a document may contain diverse information which may or may not be related to the main topic [58]. Hence more effective summarization methods are required to merge information stored in different documents and store only information related to the main topic. Many supervised and un-supervised techniques have been devised to extract salient information only [58]. Clustering based approaches like $k - means$, $hierarchical$ clustering and $topicbased$ clustering also contributed well in clustering similar sentences together in a cluster, where a sentence selection algorithm is applied and important sentences are extracted to form a summary. The basic purpose is to combine semantically similar or related sentences are put together in such an order that they share same topics [10].

The focus of all these techniques is to find a subset of the original documents in such a way where that subset is contained with core essence of documents from conceptual and semantic standpoints. In document summarization, a large feature set is a challenge that should be dealt appropriately for better quality and performance of summarization. This problem has started an enthusiasm to use Latent Semantic Analysis (LSA) to tackle curse of dimensionality. LSA is an unsupervised and most prominent tool in Information Retrieval tasks. It reveals unseen structure of documents using Singular Value Decomposition (SVD). SVD produces word-word, word-document, document-document measures that are correlated with human cognitive phenomena involving semantic or association similarity [62].

1.2 Motivation

Many automated summarization methods have been presented to deal with this increasing number of information [13, 19, 23, 32]. The prime focus is the coveying of the important ideas by reducing less crucial and redundant information. The multi document summarization is very important in dealing with a massive collection of data set. A summary contains important information from multiple documents which gives users an insight to understand a large collection of information data set in a very less time due to its compactness and coherency.

The multi document summarization can be categorised as either extractive or abstractive summarization. Extractive summarization re-arranges sentences and produces summary by highlighting important sentences, while abstractive summarization involves paraphrasing the corpus using novel sentences. Summarization methods vary depending upon its type. From feature based approaches to clustering; machine learning approaches and graph based approaches, LSA and LDA are important summarization methods [24]. Summarization is very important in today's era where information is growing rapidly with a fast pace, and we need only useful and coherent information from this large collection of data.

1.3 Problem Statement

The major problems to deal in multi document summarization is redundancy, removal of irrelevant information and producing a coherent summary. These problems were not successfully addressed in previous approaches. The resultant summary still contains irrelevant and repeated information. In previous approaches, topic detection is another research area which is is not combined with summarization. To the best of our knowledge, no such summarization approach exists which deals with the identification of the important topics that are discussed in the summarization. Readers do not know what this summary contains and which type of information it has. Furthermore, ollowing issues occur while summarizing the text documents.

1.3.1 Repetition of letters

Repeated letters, i.e, '??!!!' can be frequently observed in the text documents, which should be addressed while summarizing the documents [49].

1.3.2 Increased feature space

As a result of the continuous growth of the world wide web [59], the feature space is increasing which results in computational overhead. Size of the feature space effects the quality of the summary as irrelevant sentences or information is included in the resultant summary [64].

1.3.3 Summary about a central topic

Text documents include information about different topics, whereas, a coherent summary is usually about a central or main topic, i.e, providing information about a main idea. The sentences are semantically related and are mostly about one topic, sharing information about it from different perspectives [32].

1.3.4 Contributing topic terms

Document may have different kind of information in it, but one or two topic terms might be interesting and could have been the focused point of the summary. The knowledge of the most important topic terms in the summary would be worthy enough to share with the users beforehand.

1.4 Research Objectives

The objectives of this research is:
To preprocess the text documents to remove noise and irrelevanat informa-

tion, which increases effeciency of the further processing.
To remove redundancy from the summary, so that it contains unique unrepeated information.
To provide a coherent summary sharing an event or topic of common interest, where sentences share information about a central topic and sharing semantic relation with each others.
To present a topic to the readers which gives them an insight about the information shared in the summary.

1.5 Thesis Outline

The rest of the thesis is organized as follows: Chapter 2 provides an overview of the related state-of-the art techniques. Chapter 3 provides a detailed description of the data set and proposed framework. Following subsections in the chapter also highlight the implementation details. Chapter 4 provides comprehensive details about the evaluation measures and an experimental setup to validate our proposed framework. Chapter 5 concludes the thesis and gives insights about the future work.

Chapter 2

Literature Review

Summary contains important information from one or more documents. The size of the resultant summary is almost the half of the original text [48]. Presently, there are number of studies in the field of extractive summarization. In this section we briefly review the related work in this field, including single document summarization and multi-document summarization.
Different types of summarizers based on various factors are shown in Table 2.1.

Table 2.1: Different types of summarization systems

Summarizer types	Factors
Single vs Multi document	Size and topics of documents
Query vs Generic	Depends upon user's demand
Email based	Summarizes emails
Based on webpages	Summarizes webpages
Sentiment-based	Summarizes opinions and sentiments

A summary can be generated from one or multiple documents, which can either be generic or query-focused [40]. Emails, webpages and sentiments can also be summarized as shown in Table 2.1. Different types of summarization techniques are observed over time, we will discuss the approaches directly or indirectly related to our approach.

2.1 Single Document Summarization

Initial research in the field started with the summarization of a single document.In 1958 Luhn [34], created the first single document summarization system that uses each word frequency to measure its significance. Hence all words with significant high frequency are extracted and thus a summary of the given document is generated. However, the semantic relation between sentences is not considered in this approach, therefore the summary did not contain meaning and link with previous sentences is also not present.

Authors in [38] proposed techniques for automatic book summarization. These techniques are difficult to apply because they are not effective in cases other than book summarization.

In summarization, mainly two approaches are focused: summarization through *Supervised learning* and *Unsupervised learning.* [38]

In [29] the authors have identified the difficulty to summarize short story documents. Authors in [29] have proposed a system by combining machine learning approaches with manual assistance and achieved 6% compression ratio in summarizing short story documents. Authors have achieved good results but it is still an unsolved problem to propose an improved summarization system for short story documents that also shows improvements in compression ratio.

2.2 Feature based approaches

Authors have proposed a feature based approach, where different features like title, sentence position and its length are scored, the sentences having highest scores are selected to generate a summary [24]. A generic model for feature based approach is:

$$Score = \sum_{i=1}^{n} wi * fi, \tag{2.1}$$

where w_i = weight of feature i, f_i = score of feature i

One problem with these approaches is that they only consider the feature score of sentences not the semantic meanings between them.

2.3 Machine learning approaches

Supervised learning Machine learning approach can be applied if the training data is available. In machine learning approaches the model is trained to determine whether the element belongs to class or not. In case of summarization, model distinguishes among the sentence based upon features whether it should belong to "reference summary" or not. [43].
In machine learning approaches, a classifier is trained on a set of features. These features can be:

Table 2.2: Machine learning trainable features

Serial no	Features
1	Sentence Length
2	Position of a sentence
3	Resemblance to the title
4	Similarity to keywords
5	Occurrence of proper names
6	Occurrence of anaphors

These features are briefly described in the following subsections.

Sentence Length

This feature is set to truncate sentences which are too short, since they are not expected to belong to the summary. We use normalized length of sentence, that is the ratio of the number of words occurring in the sentences divided by the number of words occurring in the longest sentence of docu-

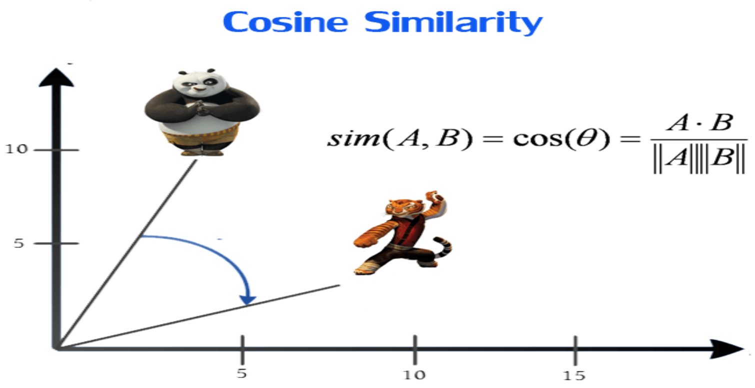


Figure 2.1: Cosine Metric for computing similarity to title [9]

ment [43].

Sentence Position

Sentence position is a combination of several items, i.e, the position of sentence in a document as a whole, with reference to a section or a particular paragraph etc.

Similarity to Title

In this feature, the similarity is calculated between setences and title. If the sentence contains title, it is important, similarity is calculated using cosine metric shown in figure 2.1.

Similarity to Keywords

This feature is obtained analogously to the previous one, considering the similarity between the set of keywords and sentences, according to the cosine similarity.

Occurrence of proper names

This feature is obtained by identifying the occurrence of proper names referring to people and places. This is considered as a binary feature with Boolean values true or false, whether a sentence contains proper name or not which is identified with the help of a part-of-speech tagger.

Occurrence of anaphors

The motivation for this feature is that the occurrence of anaphors in a sentence makes it relevant for the summary.
The classifier is trained upon above mentioned features. Correct summary sentences belonging to such categories are characterized as positive and added into the summary. The sentences which do not possess such features are labelled as negative and are automatically removed. Different classifiers can be used for training, i.e, Naive Bayes or decision tree, Support Vector Machines etc, which we have discussed briefly in the next section.
Supervised or Machine Learning approaches basically have two phases; in the first phase a classifier is trained while in the next stage testing is performed [43]. The computational complexity of O(n) sets limitation to such approaches and the un-availability of training data is another constraint.

Following table shows the machine learning/supervised approaches, due to the space constraints, we have just mentioned these approaches.

Table 2.3: Machine learning approaches

Supervised Techniques	**Methodology**
K-Nearest Neighbour (KNN)	Distance-weighted function
Support Vector Machines	Outlier detection-in hyperplane (Kernel trick)
Neural Networks	Layered (input, output)approach

2.4 Multi-document Summarization

Extraction of important information from multiple sources gained attention when some web based clustering systems were inspired by research on news like *Google News* , or *News in Essence.* [11]
This field was pioneered by authors from Columbia University in 1995 [36]. The focus in Multi document summarization is to generate summaries from multiple documents and removal redundancy, later the summarization techniques moved towards different clustering techniques using K-meand and Hierarchical clustering [28]. In these clustering techniques, the aim is to cluster similar documents together and then applying a sentence selection algorithm that can either be term frequency or sentence extraction. Authors in [65] proposed a topic basedd clustering approach for summarizing multiple documents and controlling redundancy in documents in an effective manner. However, due to topic distribution of the novel body, this approach has to sacrifice the topic diversity to a certain context. Authors in [57] presented an approach comprised upon joint graph based model for sumamrizing events which occur at different time, and achieved good results, this simple event series is not able to deal with the complex lines in the plot.

2.4.1 Clustering based Approaches

Clustering approacches are receiving much attention for the sumamrization of multiple documents. Document clustering provides effecient browsing and navigation of the corpus. Clustering can be mainly categorized into: *flat clustering* or partitional and *agglomerative clustering.* Both approaches have been investigated by researchers resulting in to *K-means, DBscan* clustering. [61]. Authors in [65] presented a probabilistic generative model where the documents are clustered and sentences in clusters are scored, the highly scored sentences are put together to form a summary. Hierarchical clustering uses bottom-up clustering to cluster data points resulting in a single big cluster. The complexity of this clustering algorithm is: $O(n^2 logn)$, where n is no of data points. Computational overhead is a barried when it comes to deal with a large number of documents. Partitional clustering, on the other hand, operates upon a pre-defining criteria of functions like in *K-means* the clusters are defined at the beginning. [61]

2.4.2 Summarization using Latent Semantic Analysis

Latent semantic indexing(LSI) or Latent semantic analysis (LSA) is used to find semantically similar terms and to reduce size of the feature space by minimizing the number of dimensions [19]
LSA requires two steps for the document representation, in the first step, the creation of a term by sentence matrix is done where columns represent the term-frequency vector of a sentence. The second step is to apply Singular Value Decomposition (SVD). to matrix A:

$$A_k = U_k \Sigma_k V_k^t. \tag{2.2}$$

SVD derives the latent structure of the document represented by the matrix A: i.e, breaks the original matrix into linearly independent vectors, which represent the main 'topics' of the document. SVD captures interrelations between terms, so terms and sentences which are arranged together will be sharing semantic meaning rather than the words only [54]. As demonstrated in [53], if a word combination is salient in a document, this pattern will be represented by one of the singular vectors, where the magnitude of the corresponding singular value shows the important degree of this pattern within the documents. Any sentences having this word combination pattern will be projected along the singular vector [14]. Authors in [21] presented an approach where standard IR methods are usedd for ranking sentences and then *Latent semantic indexing* is used to find semantic relation between sentences in order to identify important sentences. Authors in [19] proposed a technique which uses few features, adding other features like location, time and linguistic features amy further improve the results.

2.4.3 Graph based Approaches

The main idea behind this approach is to create connections between objects. A graph is generally denotes as: $G = (V, E)$, where V denotes a *vertex* and E denotes an *edge*. In context of document summarization, *vertex* represents sentences and *edge* denotes weight between two sentences, where similarity between sentences is calculated through *cosine measure.* [67]. The well known algorithms for graph based approaches are *HITS* [31] and *GooglePage* rank [45], *LexRank* [45] and *TextRank* [39] are successful raking systems that implement these algorithms. In these implementations, graph type is undirected where sentences are shown as nodes and similarity is depicted as edges. A recent graph based approach is proposed by Rafael Ferreira et al., where sentences are represented as vertices and the connection between sentences is computed using discourse, cosine and coreference resolution [51], which needs much computation and it uses wordNet to find discourse relations between words which is vocabulary limited [17]. Authors in [20] presented a graph based model for summarizing documents to overcome this limitation adding positional information to the nodes. This model produces coherent sumamry naturally, but its much focus is on the surface order of the words, which results in grouping of sentences at a surface level. The deep level demantic level grouping is not done, which van be done by overlaying parse trees.

2.4.4 Topic Modeling Summarization

The theory behind topic based summarization is to group documents as a cluster of topic words [6]. A lot of research bas been done in the field of topic modeling summarization [2, 4, 65] to summarize novel documents.

Such a topic modeling based summarization system is recently proposed by Wu, Zongda et al. [60], where a latent dirichlet allocation model is used to detect topic words and cluster the sentences using these topic terms. LDA can be defined as follows:

$$Pr(w|D) = \Sigma Pr(w|w^t|D).Pr(w^t|D), \tag{2.3}$$

where: i) Pr(w|D): The probability of occurrence of word w in document D.
ii) Pr(w|w^t): Which is probability that defines relevance of w with topic corresponding to w^t .
iii) Pr(w^t|D): The frequency of occurrence of each topic w^t with relevance to document D.
The iteration in LDA algorithm starts with: 1) Obtaining a topic w^t from each document d. 2) Obtaining a word w from the word distribution of the topic w^t. 3) Repeating the process until each word w but each document D as been traversed. Our algorithm has repeated this probability of occurrence according to the size of document D.
The sentence selection algorithm is based upon a diversity function, which is composed of positive diversity (number of topic terms) in a sentence, negative diversity (number of topic terms missing from a sentence). The sentences having highest diversity scores are chosen to form a summary. The limitation comes in using LDA, which shows no evaluation of topics over time, moreover, the redundancy issue has not been resolved properly in these approaches.

2.4.5 Critical Analysis

We have discussed **Feature based approaches**, based upon certain features, a summary is generated, but one problem with these approaches is that they only consider the feature score of sentences not the semantic meanings between them. Hence the summary only contain sentences with highest scores not the coherent sentences with logically sequenced.
The problem with **Machine learning approaches** is the computational complexity and un-availability of training data. We do not have training data available every time. The computational complexity is O(n), where n is shows number of sentences.
Training time of a classifier like **Neural Network** takes too much time and in case of **Naive Bayes** the condition of independent features is not full filled every time in real world data.
In **clustering based approaches**, two main clustering approaches are con-

sidered, *K-mean* and *Hierarchical clustering.* Former algorithm needs estimated clusters at beginning while in the later algorithm in case of bottom-up clustering, the computational overhead is a barrier when it comes to a large number of documents [61].

The **Graph based approaches** produce a coherent summary but the focus is the surface semantic level of the words instead of deep semantic level.

The **Maximal marginal relevance** is an approach for summarizing documents that matches use- less words in the query given by user and results in a relevant summary. The MMR works best for query focused summarization. However query focused documents do not give us an overview of the whole document.

The SUMMARIST text summarizer from the university of Southern California strives to produce a summary according to the following equation:

$$Summarization = topicidentification + interpretation + generation \tag{2.4}$$

, The identification of the important sentences is done at the stages *identification* and *interpretation*, which is followed by the clustering into encompassing concepts respectively. Finally summary is generated at the *generation* step, which is based upon portions of interpretation concepts. However this generation approach was not realized in the paper.

In order to deal with these limitations, we present a system that uses Latent semantic indexing and Singular value decomposition for generating only informative document matrix.

Our summarizer produces a coherent summary at deep semantic level by eliminating un necessary data and clustering similar documents together with least computational overhead. Table 2.4, shows the research gap in previous approaches.

Table 2.4: Comparision of previous research approaches

Approaches	Preprocessing	Semantic similarity	Redundancy
Zongda Wu a 2017 [13]	✓	×	×
Rafael Ferreira 2017 [19]	✓	✓	×
Xiaojun Wan 2014 [32]	✓	×	✓
Gunes Erkan 2015 [22]	✓	×	✓
Elena Baralis 2016 [67]	✓	✓	×
Yihong Gong 2013 [18]	✓	✓	×
Shengbo Guo 2013 [5]	✓	✓	×
Rasim M. Alguliyev, 2010 [20]	✓	✓	✓
Anil K. Jain 2009 [1]	✓	×	✓
KHOSROW KAIKHAH 2008 [3]	✓	×	✓
John M. Conroy 2007 [10]	✓	✓	×
Xin Liu 2005 [25]	✓	×	✓
Dingding Wang 2004 [16]	✓	×	✓
Yihong Gong 2002 [42]	✓	×	✓
Conroy Zhu 2002 [9]	✓	✓	×

Chapter 3

The Proposed Framework

This chapter is organized as follows: Section 3.1 provides the description of our data set. Section 3.2 gives an overview of our proposed framework with the detailed description of each module in the subsequent sections. Section 3.9 presents the algorithm for our proposed framework.

3.1 Data Acquisition

To perform this study, the data set we have used is provided by the National Institute of Standards and Technology (NIST). NIST is a physical sciences laboratory, and a non-regulatory agency which aims to promote research and innovation in information retrieval and industrial competitiveness.

3.1.1 Data Description

In the area of text summarization, NIST started an evaluation series, which is tentatively called Document Understanding Conference (DUC). Different DUC data sets, i.e, DUC 2001, 2002 and 2004 are available. The data set which we have used is the DUC 2004, which is also used by other state-of-the art techniques.
There are three tasks defined in each of the DUC data set, as elaborated in the Table 3.1.

Table 3.1: Specification of TASKS in DUC 2004 data set

Tasks	no. of documents	Length of words	frequency of use
Task-1	77	100	11
Task-2	59	50	4
Task-3	68	200	2

Due to its specification, we have selected TASK 1 from DUC 2004 data set which contains 77 documents on different subjects.

3.2 Proposed Framework

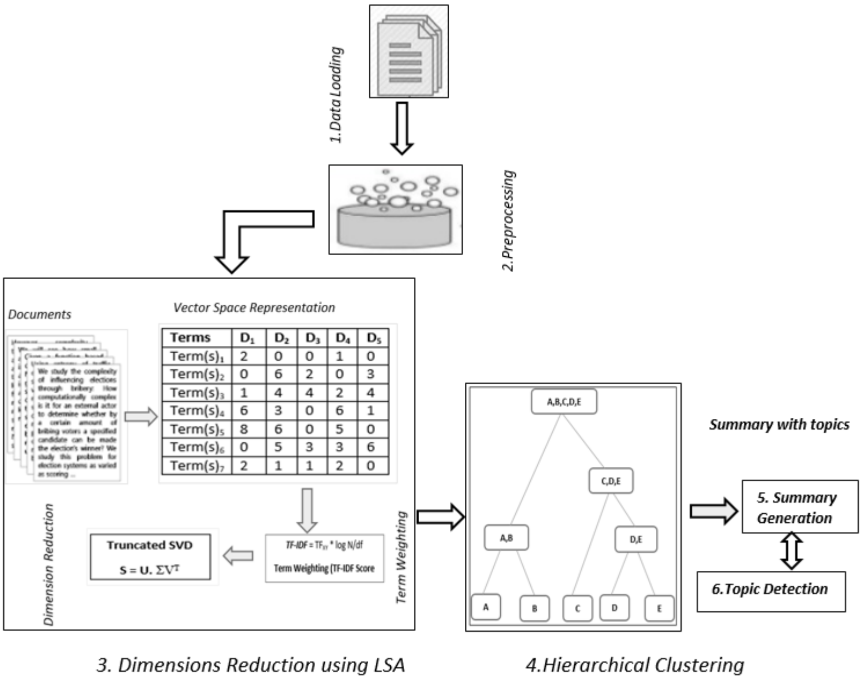
Our proposed framework consists upon six important modules. These modules are: data loading, pre-processing, dimensions reduction, clustering, summary generation and topic detection. The proposed framework for summary generation is shown in figure. 3.1, which is composed of different modules, in the following sections we will discuss each module step by step.

3.3 Data Loading

The first step in the proposed framework is data loading. We have given 77 documents from TASK 1 as an input to the system.

3.4 Data Pre-processing

In Natural language processing, Pre-processing is a salient step which enhances the quality of data by reducing the inconsistencies, incompleteness and removing noisy data [27,66]. The aim of preprocessing is to enhance the effectiveness and effeciency of the proposed approach as low quality data set generates poor summary results.
Following steps constitute data preprocessing:



We study the complexity of influencing elections through bribery: How computationally complex is it for an external actor to determine whether by a certain amount of bribing voters a specified candidate can be made the election's winner? We study this problem for election systems as varied as scoring ...

Terms	D_1	D_2	D_3	D_4	D_5
Term(s)$_1$	2	0	0	1	0
Term(s)$_2$	0	6	2	0	3
Term(s)$_3$	1	4	4	2	4
Term(s)$_4$	6	3	0	6	1
Term(s)$_5$	8	6	0	5	0
Term(s)$_6$	0	5	3	3	6
Term(s)$_7$	2	1	1	2	0

Figure 3.1: Proposed System Model for summary generation

3.4.1 Tokenization

The process of converting text into small segments is called Tokenization, these small segments are called tokens. The purpose of tokenization is to examine words and feed these words as an input for further operations, i.e, stop words and punctuation removal. Tokenization is not a simple step as it may contain a sequence of alphabetic, non-alphabetic or alphanumeric characters. Like alphabetic sequences, alphanumeric should also be treated as a single token [12,55]. In order to perform tokenization, a scikit-library is used in our proposed approach. The pseudo code for tokenization is shown below:

3.4.2 Stop Words Removal

Frequently repeated terms in documents are called stop words, which occur without any particular meaning or relation to the topic, i.e, conjunctions, prepositions etc. Stop words are ignored and removed from the sentences

Algorithm 1: Algorithm for tokenizing the sentences

```
Input  : A set of documents, doc_set
Output: A set of tokens
for i in doc_set do
    raw = i.lower()
    tokens = tokenizer.tokenize(raw)
end
 return tokens
```

because they, (1) do not carry useful and discriminative information, (ii) create hurdles in further processing and understanding of the contents, (iii) give weight to terms due to their frequent occurrences, which have no importance or meaning [37]. Stop words are removed to improve text quality and reducing feature space [47].

In our proposed approach, we have used a Python library called Natural language toolkit (NLTK) to get a list of stopwords [44].

The pseudo code to remove stop words is shown below in algorithm 2:

Algorithm 2: Algorithm for removing stopwords

```
Input  : A set tokens, tokens
Output: Set of Clean_tokens without stop words
en-stop = set(stopwords.words('english'))
stopped-tokens = i for for i in tokens do
    if not i in en-stop then
        return clean_tokens
    end
end
```

3.4.3 Punctuation Removal

Our choice of including punctuation (. , : ; ? ! - etc) removal is dependent upon both what the model does and how we intend to use the word embeddings generated by the model. Especially for Doc2BOW model, which learns embedding without considering order of the words, thus making it

an essential preprocessing step. Punctuation provides grammatical context which supports understanding of the content but not in case of the vector space representation [26].
The pseudo code for punctuation removal is shown below.

Algorithm 3: Algorithm for removing punctuations

```
Input  : A set of documents
Output: A set of data $D$ without punctuations
for char in txt: do
    if char not in punctuations: then
        no_punct = no_punct + char
        return $D$
    end
end
```

After preprocessing, our data is free of noise and raw facts and figures. The preprocessing does effect the feature space as shown in Table 3.2, as the preprocessing steps include removal of frequent stop words, punctuations, which results in a reduced feature space.

Table 3.2: Effects of preprocessing on Bag of Words feature space

Preprocessing parameters	**Size**
Feature space without pre-processing	21,599
Feature space after Stop words removal	17,678
Feature space after Punctuation removal	14,578
Feature space after all preprocessing steps	10,3427

3.5 Dimensions reduction using Latent Semantic Analysis (LSA)

Documents may contain terms which are semantically identical, hence result in increased size of the feature space. LSA is a technique which compares text using a vector space representation which is learned from a corpus. The primarily task performed by LSA is to compute the similarity between pair of text [8]. The following subsection provides the steps performed in our proposed framework to reduce the dimensions.

3.5.1 Vector Space Representation

The first step towards dimension reduction is the vector space representation of the text documents.

Vector space representation is an algebraic model for representing text documents as a set of vector identifiers, such as index terms [15]. We have used bag of word (doc2BOW) model for vector space representation provided by gensim. Gensim is a robust open-source vector space modeling toolkit implemented in Python. Gensim includes implementation of word2vec, $tfidf$, latent semantic analysis and doc2bow models.

The doc2BOW model works as follows:

1. In the first step, a dictionary is created with unique terms

2. The next step is to create a document term matrix, which contains two elements (1) term id (2) term frequency. The terms in the dictionary are used to create a document term matrix. The code for creating term matrix is shown in algorithm 4.

Algorithm 4: Algorithm for creating Document term matrix

```
Input   : Document corpus
Output: Document term matrix, $D$
dictionary = corpora.Dictionary(texts)
corpus = dictionary.doc2bow(text)
for text in texts do
    return $D$
end
```

The document term matrix provides us information about the occurrences of the terms in corpus. This term matrix is given as an input to our next step LSA.

3.5.2 Singular Value Decomposition (SVD)

LSA extends the vector space representation using Singular Value Decomposition (SVD) to reconfigure data. SVD is an algebraic technique which re-orients and ranks the dimensions in a vector space [54]. SVD is viewed as a best approximation of the original data set using fewer dimensions. Hence, SVD can be identified as a tool for data or dimension reduction. The idea that makes SVD particular for Natural language processing (NLP) applications is that we can simply ignore variation below a particular threshold, while relation of interest between original data has been assured [30]. The algorithm of this step is shown below:

Algorithm 5: Algorithm for reducing dimensions

```
Input  : Document Term Matrix
Output: A reduced set of terms, $RD$
tfidf = models.TfidfModel(corp) corpus-tfidf = tfidf[corp]
for $document\ in\ corpus - tfidf$ do
    lsi = models.lsimodel.LsiModel(corpus-tfidf, id2word=dictionary,
      num-topics=50) return $RD$
end
```

LSA needs term frequency Inverse document frequency of the the terms. $Tfidf$ calculates the frequency of the terms in the entire corpora. It is defined in Section 1. This $tfidf$ model works as an input to LSA for further processing. Three parameters are specified while calling Lsa function. The dictionary created in the previous step, the parameter to create num topics and the tfidf matrix. LSA or LSI (latent semantic indexing) is provided in gensim library and can be accessed using a single command as shown in algorithm. At the end of this step we have a reduced set of dimensions.

3.6 Hierarchical Agglomerative Clustering (HAC)

LSA provides us a reduced set of dimensions, to create a relationship among these terms, this section provides the implementation of Hierarchical agglomerative clustering algorithm. We need a summary, where sentences share semantic meanings and are related to a main topic. Instead of sharing different information, a summary should be coherent to a main topic. For having different topics, different number of summaries are generated. In order to cluster terms together, we have applied HAC on our set of terms.

3.6.1 Bottom-up approach

In our proposed system we have used Bottom-up approach where each node in a cluster contains a group of similar data clusters at one level join to other clusters in next level up, using a similarity degree. This process continues until all nodes form up a single cluster. In HAC the total number of clusters is not predetermined. Popular options of linkage in HAC are:

Complete linkage: It is the similarity of the farthest pair. One drawback of the complete linkage is that outliers can cause merging of close groups later than is optimal.

Single Linkage: It is measure of similarity of closet pairs, this can cause premature merging of groups even if they are dissimilar to each other.

Group average: similarity between groups [41]. Cluster distance between points c_1, c_2 is calculated as:

$$D(c_1, c_2) = \frac{1}{|c_1|}\frac{1}{|c_2|}\Sigma_{x_1 \epsilon c_1}\Sigma_{x_2 \epsilon c_2} D(x_1, x_2),$$

where c_1, c_2 denote clusters, while x_1, x_2 show data points [41].

In our system, we have set following parameters:

i) Algorithm is Agglomerative; ii) linkage is average. We have cut our tree at level two and level one for two different data sets which we will describe in experimental section in detail. Average link calculates average of similarity between all inter-cluster pairs. At the end of this step, we have clusters consist of related terms on which we can apply our selection algorithm to select sentences for our summary. All related terms are now contained in one cluster whose computational complexity is also decreased as we have already semantically related terms with highest eigen values using LSA. Clustering combines the similar terms together in a group. Then we apply our sentence selection algorithm. The pesudo code of our clustering is shown below. HAC needs different parameters, i.e, the linkage which in our case is average, and the term matrix. We have cut the tree at level three, where maximum similarity between clusters is shown, in our case we have got two clusters, the terms within clusters are semantically related.

```
Algorithm 6: Algorithm for Hierarchical agglomerative clustering
  Input  : Document Term Matrix by LSA, $doc\_feat$
  Output: Number of clusters having terms, $clusters$
  model = $AgglomerativeClustering(linkage =' average', connectivity = None) model.fit(\text{doc\_feat}.toarray())$
  Z = $cluster.hierarchy.ward(X)$ clusters = $cluster.hierarchy.cut\_tree(3)$
  return $clusters$
```

3.7 Summary generation

In this section, we have discussed our next module which is summary generation.

3.7.1 Sentence selection based upon tfidf scores

Our sentence selecction algorithm is based upon $tfidf$. We have extracted the sentences which have highest weighted terms defined by *tfidf* in them, to form a summary.

At the end of this step, we have a coherent summary from multiple documents. The psuedo code is shown below:

Algorithm 7: Algrothm for extracting sentences with *tfidf*

```
Input  : Set of clustered terms, docu
Output: Sentences having high tfidf scores, S
vectorizer = TfidfVectorizer() X = vectorizer.fit_transform(docu)
 indices = np.argsort(vectorizer.idf_)[::-1] features =
 vectorizer.get_feature_names() top_n = 15 top_features =
 [features[i] for i in indices[:top_n] do
    sentences in doc_set
end
file = list.read().split() for word in arr: do
    for linenum, line in enumerate(top_n): do
        if word in line.lower(): then
            w = open('summary.txt', 'w', encoding="utf8")
             w.write(str(line))
        end
    end
end
return S
```

As the algorithm depicts, we have applied *tfidf* vectorizer upon the clustered terms and selected the top feature sets which have higher *tfidf* weight, then we have selected the sentences having these *tfidf* terms.

3.8 Topic Detection

In our proposed model, the topic related to summary is also provided to user after summary is produced.

3.8.1 Latent dirichlet Allocation Model (LDA)

We are using Latent Dirichlet Allocation(LDA) for topic modeling. LDA is a generic probabilistic model. It automatically discovers *Topics* that these sentences contain. LDA decides Topics on the basis of poison distribution which is popular for modeling the number of times an event occurs in interval of time and space. We have used the Genism tool [50] to carry out LDA topic modeling.

Algorithm 8: Algorithm for detecting most contributing topic terms from sumamry

```
Input  : Summary, $S$
Output: Most contributing topic terms, $T\_n$
corpus = [dictionary.doc2bow(text) for text in texts]
ldamodel = gensim.models.ldamodel.LdaModel(corpus,
  $num\_topics$=10, id2word=dictionary, passes=20) $T\_n$ =
  ldamodel.$print\_topics$()
return $T\_n$
```

The summary is given as an input to the LDA model, which coverts the text into vectors using doc2BOW model. The parameters LDA requires, beside the text corpus are, the number of topics, the dictionary created by doc2BOW and the total number of runs, i.e, LDA iterates this number of passes to generate the most contributing topic terms.

3.9 Algorithm for Proposed framework

The algorithm for our proposed system model is depicted below which depicts overview of each module discussed in the above sections.

Algorithm 4 depicts overview of the modules, while the detail of each step is provided in the previous subsection of modules. The above algorithm starts with browsing documents for summarization. In step 2 preprocessing, steps 6-9 contain Vector space modeling using doc2BOW to represent documents as a set of vectors and dimensions reduction using LSA respectively, steps 10-13, show clustering algorithm, steps 15-19 show $tfidf$ steps to select sentences and step 20 shows topic detection using LDA.

Algorithm 9: Algorithm for the proposed framework

```
Input   : Set of Documents $N$
Output: Summary S with Topic Terms $T_m$
Step 1. Browse documents N
Step 2. Preprocessing
            Step 2.$a_1$. Tokenization
            Step 2.$a_2$. Stop words removal
            Step 2.$a_3$. Punctuation removal
Step 3. Build Count Matrix M
 corpus = [dictionary.doc2bow(text) for text in texts]
Step 4. Apply LSA Model
     models.lsamodel.LsaModel(corpus$_t fidf$, id2word=dictionary,
  num$_t$opics=2)
Step 5. Apply Agglomerative Hierarchical Clustering
     model = Agglomerative Clustering(linkage='average',
     connectivity=None, $n_c lusters = clusters$)
     $D(c_1,c_2)= \frac{1}{|c_1|}\frac{1}{|c_2|}\Sigma_{x_1 \epsilon c_1}\ \Sigma_{x_2 \epsilon c_2} D(x_1,x_2)$

Step 6. Apply sentence selection algorithm
     tfidf(t,d).idf(t,D)
while all sentences having MAXTERM do
  | add sentence into summary S
end
Step 7. Apply LDA to generate topic terms $T_n$ on Summary $S$, ldamodel
  = gensim.models.ldamodel.LdaModel(corpus, num_topics=10
  id2word=dictionary, passes=20) T_n = ldamodel.print_topics()
```

Chapter 4

Experiments and Evaluation

This chapter is devoted to describe the experiments conducted to evaluate our proposed framework. A total of two data sets are used for the evaluation purpose. In this chapter we have discussed about the evaluation metrics in the first section. The next section is data specification, which describes the data set, we have used in this evaluation measure. Two different experimental setups are performed for evaluation on two different data sets in section three and four.

4.1 Evaluation Metrics

We have used Recall-Oriented Understudy for Gisting Evaluation (ROUGE) [33] as our evaluation metric. ROUGE is a famous metric which is widely used by DUC as a standard for evaluating automatic summaries. ROUGE measures the unit overlaps between candidate and a reference summary. ROUGE provides several implementations, i.e, ROUGE-SU, ROUGE-W, ROUGE-L, ROUGE-N. ROUGE-N which is n-gram recall measure, computed as:

$$ROUGE-N = \frac{\sum\limits_{S\in\{RefSum\}} \sum\limits_{n-gram\in S} Count_{match}(n-gram)}{\sum\limits_{S\in\{RefSum\}} \sum\limits_{n-gram\in S} Count(n-gram)}, \qquad (4.1)$$

Where ref is for reference summaries while n is for length of n-gram.

Count match($gram_n$) is the count of maximum number of n-grams occurring in both reference and candidate summaries. ROUGE-SU is a measure of skip plus uni-gram. ROUGE-L matches the longest common subsequence (LCS). ROUGE-W is weighted is weighted LCS [52]. Each of these ROUGE methods provide *Precision, Recall* and *f-measure* scores, which are calculated as follows:

$$Precision(D, M) = \frac{1}{|S|} \sum_{S_k \in S} \frac{|S_k^d \cap S_k^m|}{|S_k^d|}, \tag{4.2}$$

Where: D is defined as our algorithm and $\mathbb{M}$ as a summary
S_k^d, S_k^m denote the automatic summary generated by our algorithm, and manual summary respectively. The quality is calculated by computing intersection.

The Recall and f-score is computed as follows:

$$Recall(D, M) = \frac{1}{|S|} \sum_{S_k \in S} \frac{|S_k^d \cap S_k^m|}{|S_k^m|}. \tag{4.3}$$

$$F1 = \frac{2 \cdot precision \cdot recall}{precision + recall}. \tag{4.4}$$

4.2 Data Specification

We have used Document Understanding Conference (DUC) data set 2004 provided by National Institute of Standards and Technology (NIST), the more details about this data set can be found in Section 3.1.1. We have conducted two experiments, the first experiment is conducted on DUC 2004 data set TASK-1. This data set contains 77 documents about different subjects. The second experiment is conducted on a data set provided by the institute research group.

The subsequent sections depict the results of our proposed framework on these data sets.

4.3 Experiment 1

The first data set provided by NIST is DUC data set 2004. We have used TASK1 which contains 77 different documents. In the following steps, we show the result of each module using our proposed framework, due to space constraints and large number of data sset, we will show only smaple results in experiment 1 while experiment 2 contains detailed results of each module. In the first phase, we have loaded 77 documents in our system. The second step contains preprocessing, results of each module are shown in the following subsections. The sample of the documents is shown in the figure. REF

4.3.1 Sample of data set

Damascus 10-18 (AFP) - A decree issued by Syrian President Hafez Al-Assad today, Sunday, announced that the legislative elections will be held next November 30 to elect new members in the People's Congress (the Parliament).

The decree explained that representatives from 15 Syrian governances must be elected to occupy the congress seats numbered at 250, including 127 belonging to ”laborers and farmers” and 123 ”to other classes of people.”This allocation is identical to the seat allocation in the Parliament, the term of which had expired last September 9.

It should be noted that the congress term in Syria is four years.It should be mentioned that during the last legislative elections in August 1994, the ruling coalition in Syria, ”The Progressive National Front”, gained 167 seats out of the 250 the congress is comprised of.All candidates of the coalition, which encompasses seven parties including the Baath Party, have been elected, while the independent representatives gained 83 seats.

Tehran 10-30 (AFP) - The Iranian Islamic Republic's spiritual guide, Ayatollah Ali Khameni, strongly denounced the Palestinian Authority President Yasser Arafat today, Friday, for signing the Wye Plantation Accord with Israel and called him ”a traitor and a follower of the Zionists.

4.3.2 Preprocessing results

This module contains important steps for preprocessing the data, i.e, tokenization, stopwords removal and punctuation removal. The sample result of each module is shown below.

4.3.3 Result sample for tokeniztaion

```
[’damascus’, ’10-18’, ’(afp)’, ’-’, ’decree’, ’issued’, ’syrian’, ’president’, ’hafez’, ’al-assad’, ’today,’, ’sunday,’, ’announced’, ’legislative’, ’elections’, ’held’, ’next’, ’november’, ’30’, ’elect’, ’new’, ’members’, ”people’s”, ’congress’, ’(the’, ’parliament).the’, ’decree’, ’explained’, ’representatives’, ’15’, ’syrian’, ’governances’, ’must’, ’elected’, ’occupy’, ’congress’, ’seats’, ’numbered’, ’250,’, ’including’, ’127’, ’belonging’, ’”laborers’, ’farmers”’, ’123’, ’”to’, ’classes’, ’people.”this’, ’allocation’, ’identical’, ’seat’, ’allocation’, ’parliament,’, ’term’, ’expired’, ’last’, ’september’, ’9.’, ’noted’, ’congress’, ’term’, ’syria’, ’four’, ’years.it’, ’mentioned’, ’last’, ’legislative’, ’elections’, ’august’, ’1994,’, ’ruling’, ’coalition’, ’syria,’, ’”the’, ’progressive’, ’national’, ’front”,’, ’gained’, ’167’, ’seats’, ’250’, ’congress’, ’comprised’, ’of.all’, ’candidates’, ’coalition,’, ’encompasses’, ’seven’, ’parties’, ’including’, ’baath’, ’party,’, ’elected,’, ’independent’, ’representatives’, ’gained’, ’83’, ’seats.’] [’tehran’, ’10-30’, ’(afp)’, ’-’, ’iranian’, ’islamic’, ”republic’s”, ’spiritual’, ’guide,’, ’ayatollah’, ’ali’, ’khameni,’, ’strongly’, ’denounced’, ’palestinian’, ’authority’, ’president’, ’yasser’, ’arafat’, ’today,’, ’friday,’, ’signing’, ’wye’, ’plantation’, ’accord’, ’israel’, ’called’, ’”a’, ’traitor’, ’follower’, ’zionists.”’]....
```

Our first preprocessing step includes tokenization, whose result is depicted in the above sample where it breaks the sentences into tokens.

The second and third preprocessing steps include stopwords and punctuations removal. The following sample shows the result of these steps.

4.3.4 Result sample for stopwords and punctuation removal

```
damascus 1018 afp decree issued syrian president hafez alassad today sunday announced legislative elections held next november 30 elect new members peoples congress the parliamentthe decree explained representatives 15 syrian governances must elected occupy congress seats numbered 250 including 127 belonging laborers farmers 123 to classes peoplethis allocation identical seat allocation parliament term expired last september 9 noted congress term syria four yearsit mentioned last legislative elections august 1994 ruling coalition syria the progressive national front gained 167 seats 250 congress comprised ofall candidates coalition encompasses seven parties including baath party elected independent representatives gained 83 seats
tehran 1030 afp iranian islamic republics spiritual guide ayatollah ali khameni strongly denounced palestinian authority president yasser arafat today friday signing wye plantation accord israel called a traitor follower zionists algerian regimes strongman withdraws political arenaalgiers 1020 afp mohammad bushtein algerian regimes strongman withdrew political arena result campaign press him denounced excesses committed time candidate come forward presidential elections due held early 1999 president liamine zerouals successionbushtein resigned post advisor algerian administration opting embarrass government coalition effort appoint candidate replace president zeroual latter decided give post early coming year end termthe press campaign launched opposition beginning last summer season aimed affirming bushtein took advantage position counseling minister president realize private gains one hand eliminate ene mies otherpresident zeroual surprised everyone last september 11 announced abbreviation presidential term supposedly due end year two thousand planning early presidential elections candidate.....
```

Now we have applied doc2BOW model upon our data set to convert text into vectors. The first element shows the term id and the second element shows term frequency. Total number of documents are 77, which are represented by 77 vectors as shown in the result sample below.

4.3.5 Result sample of Vector space modeling

Table 4.1: Document term matrix, result sample

[[(0, 1)	(1, 1)	(2, 1)	(3, 1)	(4, 1)
(5, 1)	(6, 1)	(7, 1)	(8, 1)	(9, 1)
(10, 1)	(11, 1)	(12, 1)	(13, 1)	(14, 1)
(15, 1)	(16, 1)	(17, 1)	(18, 1)	(19, 1)
(20, 1)	(21, 1)	(22, 1)	(23, 1)	(24, 1).....]]

4.3.6 Dimension Reduction Results

In this module, we have reduced our set of dimensions using LSA. LSA uses SVD, which preserves the terms having highest variation and give us the data from which all the information about the dataset can be retrieved. LSA first applies tfidf on the given document term matrix that is produced in the previous step. Here we have shown the result sample of this module.

4.3.7 Result sample of $tfidf$

The result sample shows the tfidf of all the terms, where the first element shows the id of the term while the other term shows the tfidf weight of that term. It gives us the frequency of the terms from all documents. Upon this model, LSA model is applied which gives us the most contributing terms using SVD.

H

Table 4.2: Document term matrix, result sample

[(0, 0.19733078912534788)	(1, 0.19733078912534788)	(2, 0.1973307891288)
(3, 0.19733078912534788)	(4, 0.19733078912534788)	(5, 0.0480558097035)
(5, 0.0480s55809730828035)	(6, 0.19733078912534788)	(7, 0.19733078988)
(8, 0.19733078912534788)	(9, 0.19733078912534788)	(10, 0.197330734788)....)]

4.3.8 Result sample of LSA

```
[(1, tehran” + traitor” + ”follower” +”arafat” + ”islamic” + ”zionists”’),
( 2, ”congress” + ”seats” + ”presidential” + ”candidate” + ”algerian” +
”elections” + ”legislative” + ”representatives”’), (3, ”king” + ”altarawneh”
”chemotherapy” + ”medical” + *”treatment”’), (4,”congress” + ”allocation”
+ ”syrian” ....)]
```

LSA gives us the important terms from all 77 documents as we can see in the above sample reuslt, where the irrelevant terms are removed and only the important dimensions which are semantically related to each other are preserved in this module, i.e, treatment and medical. The terms represented by LSA are semantically related.

4.3.9 Results of clustering

LSA gives us the terms which are semantically related to one another as we have seen in previous sample result. Document-1 terms are separated from document-2 and so on. The relation between doc-1 and doc-2 terms, is checked using HAC algorithm. HAC tells us the relation between all the terms in all documents in the form of a cluster.

4.3.10 Result sample of HAC

```
[0.congress, seats, presidential, candidate, early, algerian, last, elections,
legislative, representatives.....]

[1. medical, altarawneh, chemotherapy, mayo, treatment....]
```

HAC generally provides us a single cluster as it is a bottom-up approach and all the clusters are summed up into a single cluster at the end, but it contains mostly all the terms. To find a point where maximum similarity between terms is lied, we have cut the HAC dendogram tree at level 3. This cut is done by a hit and trial method, that is achieved where we get best results. Cutting the HAC at level 3, it returns us two clusters. The clusters contain similar terms, our number of summaries depend upon the number of clusters. As two clusters show that two different topics are discussed in the data set, which are distinguished using HAC.

4.3.11 Summary generation

HAC returns us a large number of terms, all of them are may be not important. To compute the most important terms, we have applied tfidf upon these terms. The terms from both clusters, which have tfidf score greater than 0.15 are considered as important and put together.

In sentence selection step, we have looped through all the documents and selected the sentences containing $tfidf$ terms. All the sentences are put together and a coherent summary is generated.

4.3.12 Result sample of summary:1

Algerian Regime’s Strongman Withdraws from Political Arena. Algiers 10-20 (AFP) - Mohammad Bushtein, the Algerian regime’s strongman, withdrew from the political arena as a result of a campaign by the press against him, in which it denounced the ”excesses” he committed, at a time when no candidate has come forward for the presidential elections which are due to be held in early 1999 for President Liamine Zeroual’s succession. Bushtein resigned his post as an advisor to the Algerian administration opting not to embarrass the government coalition in its effort to appoint a candidate to replace President Zeroual.

damascus 1018 afp decree issued syrian president hafez alassad today sunday announced legislative elections held next november. 30 elect new members peoples congress the parliamentthe decree explained representatives 15 syrian governances must elected occupy congress seats numbered 250 including 127 belonging laborers farmers 123 to classes. peoplethis allocation identical seat allocation parliament term expired last september 9 noted congress term syria four yearsit mentioned last legislative elections august 1994 ruling coalition syria the progressive national front gained 167 seats 250 congress comprised of all candidates coalition encompasses seven parties including baath party elected independent representatives gained 83 seats.

The press campaign launched by the opposition in the beginning of last summer season was aimed at affirming that Bushtein took advantage of his position as a counseling minister to the president to realize private gains on the one hand, and to eliminate his ene mies on the other. President Zeroual had surprised everyone last September 11 when he announced the abbreviation of his presidential term, which was supposedly due to end in the year two thousand, and the planning of early presidential elections in which he will not be a candidate.

As we have two clusters containing two major topics based upon which two different summaries are created. Summary result for sumamry-2 is shown below.

4.3.13 Result sample of summary:2

’king hussein begins fourth phase chemotherapy amman 106, jordans prime minister fayez altarawneh’ said king hussein american mayo clinic hospital since mid july began fourth phase chemotherapy yesterday monday two phases left, fight lymphatic node cancerin statements press upon arrival amman yesterday evening mayo clinic rochester state minnesota,

King hussein returned altarawneh said king hussein began yesterday fourth phase treatment enjoying good health responding excellent manner intense medical supervision continuous evaluation procedures state television showed day yesterday scenes. King husseins meeting prime minister chief royal court jawad alanani foreign minister abdul elah alkhatibthe jordanian monarch expected return home first half next november eve sixtythird birthday falls 14 november altarawneh pointed the intense medical supervision one reasons delayed fourth phase several days, however, consulting large number specialists come prominent medical centers united states evaluation medical status unanimously agreed yesterday king continue phase way according treatment schedule jordanian monarch confirmed september 20 completed third six phases chemotherapy prescribed doctors mayo clinic treatment period extends four days separated one another three weeks king hussein admitted american specialist hospital suffered sweating spells rise temperatures doctors diagnosed condition lymphatic node cancer

4.3.14 Topic detection results

The last module is topic detection suing LDA. In this step, we apply LDA on our two different summaries which give us the important topic terms from both summaries. Topic detection lets the reader get an insight about the summary topic.

4.3.15 Result sample for topic detection; summary1

congress, seats, presidential, candidate,
algerian, elections, legislative, representatives

4.3.16 Result sample for topic detection; summary2

hussein, phase, fourth, medical,
altarawneh, chemotherapy, mayo, treatment

The results on topic detection module shows the important topic terms discussed in these two summaries, i.e, legislative elections, and King Hussein medical status. The readers, in prior, without reading entire collection of documents or even summaries, can get information about the topics discussed in the documents, he/she can choose according to the interest to read the summary or not. This module is really useful in terms of both, effeciency and effectivness.

4.3.17 Evaluation Results on Experiment-1

In this section, we have evaluated our summaries using ROUGE metric. We have compared our results with the state-of-the art techniques, i.e, DUC peer model; Random summarizer and TextRank. The specification of these systems is shown in the following table.

Table 4.3: Specification of the Benchmark techniques

Specifications	Dataset	Framework
DUC Peer code-1	DUC-2004 TASK1	Graph based
Random summarizer	DUC-2004 TASK1	Clustering based
TextRank	DUC-2004 TASK1	Graph based

The Textrank approach [35] and the DUC peer model are the graph based approaches, where nodes and edges are created based upon the sentences similarity. The sentence which shares most of the similarity with other sentence, is considered important and selected to form a summary [56]. The Random sumamrizer [68] randomly selects the sentences to form a summary from the clusters which are formed by fuzzy clustering [13].

4.3.18 Results on: ROUGE-1

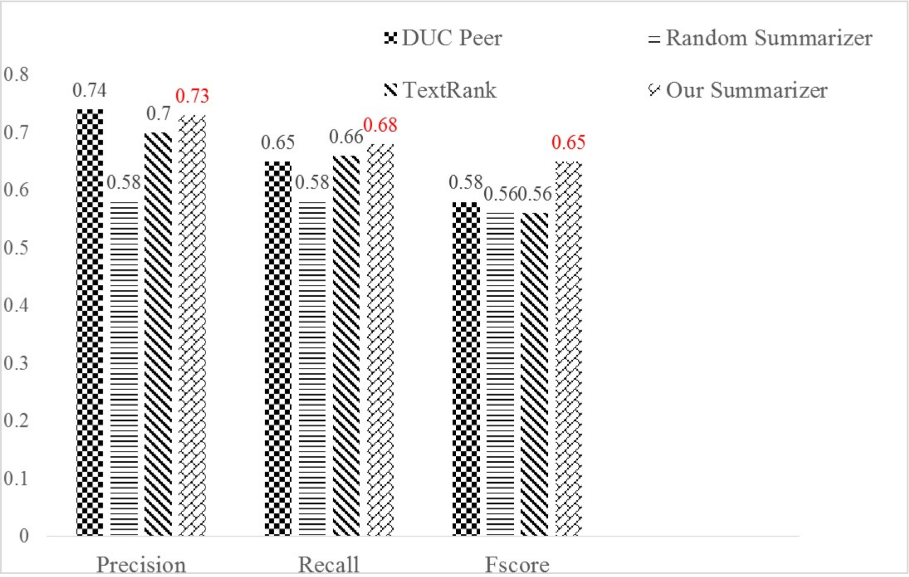


Figure 4.1: Evaluation scores on Rouge-1: dataset(DUC'04)

Figure 4.1 shows the evaluation scores on ROUGE-1, DUC data set (2004). These are the results from experiment-1. ROUGE scores result in precision, recall and fscore, ROUGE checks the overlapping between original text and the automated summaries, as discussed in section 4.1. The reslts show the highest precision and recall scores by our system, hence our system outperformed in two metrics on ROUGE-1. In the following figure, ROUGE-2 scores are shown.

4.3.19 Results on: ROUGE-2

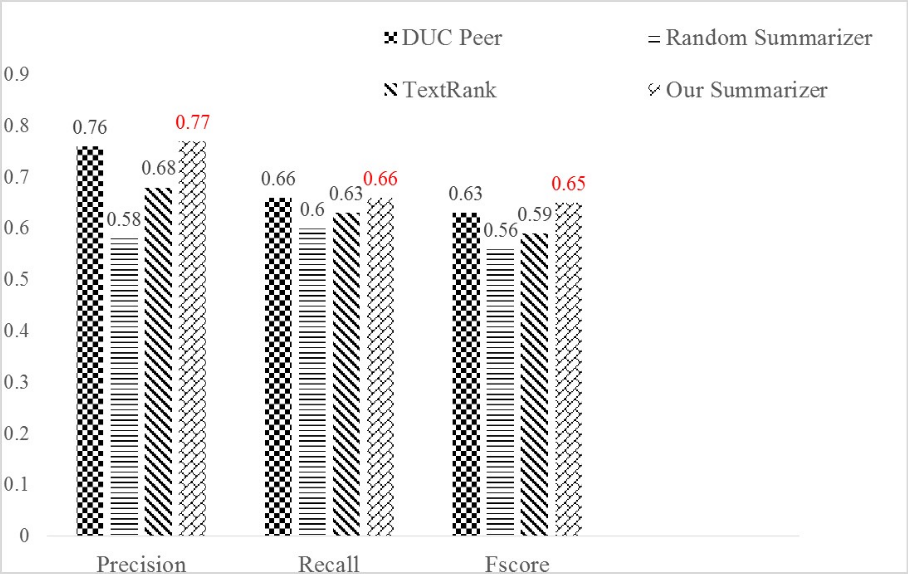


Figure 4.2: Evaluation scores on Rouge-2: dataset(DUC'04)

Figure 4.2 shows ROUGE-2 scores on DUC 2004 data set. Our System on ROUGE-2 scored between 0.6-0.77 on all three ROUGE matrices, i.e, Precision, Recall and Fscore. Our system has outperformed on two scales precision and fscore.

4.3.20 Results on: ROUGE-SU4

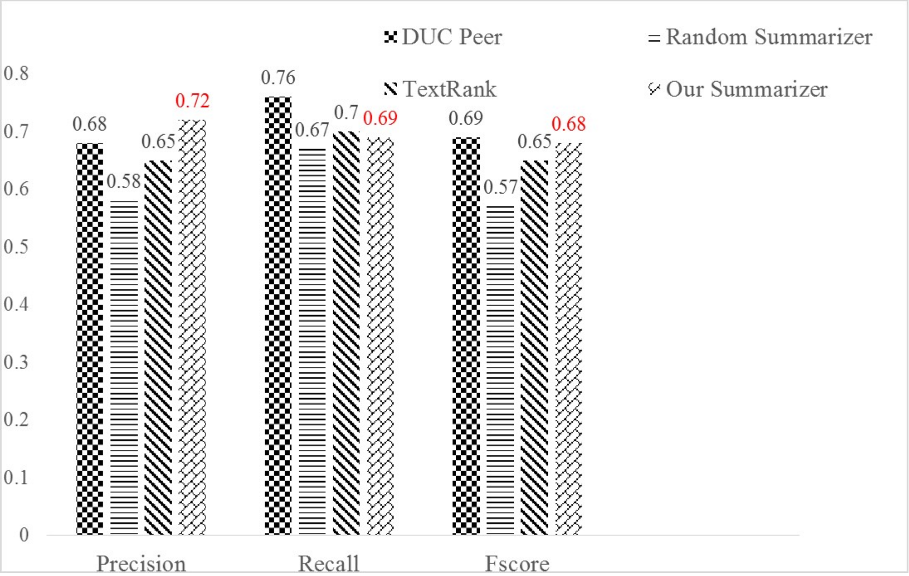


Figure 4.3: Evaluation scores on Rouge-SU4: dataset(DUC'04)

As we can see in Fig 4.3, ROUGE-SU4 score of our system lies between 0.6-0.7. The highest Recall score on ROUGESU4 is 0.72 on scale precision. In comparison with other systems on DUC data set 2004, our system outperforms in terms of precision, recall and Fscore. The evaluation scores on TASK 1 against (A-Z) peer codes, (top four submissions) are depicted in the following tables 4.5 and 4.6. In the light of all these experiment results, we conclude that our summarizer generates optimal machine summary. Ensuring high quality and effectiveness. Our system has outperformed existing techniques by over 5% on ROUGE scoring scale. It can be observed that the system based on LSA and clustering produces a coherent summary by grouping semantically related sentences and extracting highly weighted terms. Our system shows efficient results as compared to other systems which used graph based and feature based approaches. The summary is effective and human readable.

Table 4.4: Results on DUC Peer code (A-F)

Tasks	**Rouge-1**	**95% Confidence Interval**
A	0.3933	0.3722,0.4143
E	0.4104	0.3882,0.4326
F	0.4125	0.3916,0.4333
H	0.4183	0.4019,0.4036
Our Summarizer	0.5100	0.4446,0..4029

Table 4.5: Results on DUC Peer code (W-Z)

Tasks	**Rouge-1**	**95% Confidence Interval**
W	0.4119	0.3870,0.4368
X	0.4293	0.4068,0.4517
Y	0.4445	0.4230,0.4660
Z	0.4326	0.4088,0.4565
Our Summarizer	0.5100	0.4488,0..4203

4.4 Experiment 2

The second data set provided by the institute research group conatins five documents. This experiment is conducted on these five documents. These documents are shown below:

Table 4.6: Data set for experiment:2

Doc1:	Hot Chocolate Cocoa Beans
Doc2:	Beans Harvest Cocoa Butter
Doc3:	Sweet Chocolate Butter Sugar
Doc4:	Sugar Cane Ice cream Beat
Doc5:	Sweet Sugar Beat Black

4.4.1 Results on module 1: preprocessing

As the data is already pre processed (no stopwords and punctuation marks appear in this data set). We apply tokenization to make tokens.

Table 4.7: Tokenization results for experiment:2

Doc-1:	’hot’,	’chocolate’,	’cocoa’,	’beans’	
Doc-2:	’beans’,	’harvest’,	’cocoa’,	’butter’	
Doc-3:	’sweet’,	’chocolate’,	’butter’,	’sugar’	
Doc-4:	’Sugar’,	’Cane’,	’Ice’,	’cream’,	’Beat’
Doc-5:	’Sweet’,	’Sugar’,	’Beat’,	’Black’	

4.4.2 Punctuation Results on Experiment:2

Table 4.8: Punctuation results for experiment:2

Doc-1:	hot	chocolate	cocoa	beans
Doc-2:	beans	harvest	cocoa	butter
Doc-3:	sweet	chocolate	butter	sugar
Doc-4:	sugar	cane	icecream	beat
Doc-5:	sweet	sugar	beat	black

4.4.3 Vector Space Representation results on Experiment:2

The doc2BOW model is applied on the data set which creates a dictionary and a vector representation of the documents. The dictionary, in this experiment, is created with seven unique tokens:
Dictionary(7 unique tokens: [’cocoa’, ’sweet’, ’beans’, ’beat’, ’sugar’]...)
The five documents are represented by five vectors. BOW represents each token as a combination of two elements. First is the ID of the token and second is the occurrence of the terms.For example the first term which is

'cocoa' occurs once in first document, it is represented as (0,1) which shows cocoa is first element with occurrence one.

Table 4.9: Document term matrix for experiment:2

Doc-1	(0, 1)	(1, 1)	(2, 1)	
Doc-2	(0, 1)	(1, 1)	(3, 1)	
Doc-3	(2, 1)	(3, 1)	(4, 1)	(5, 1)
Doc-4	(4, 1)	(6,		
Doc-5	(4, 1)	(5, 1)	(6, 1)	

4.4.4 Results on module: Dimension Reduction

The doc2bow model doesn't give us an idea of occurrence of a term in all documents. We need another mechanism which calculates the frequency of occurrences in all documents. For this we have applied tfidf which calculates term frequency and inverse document frequency, it can give us an idea of occurrence of terms in all documents. This tfidf matrix is given as an input to the LSA model.
The results of tfidf model is shown here:

Table 4.10: tfidf matrix for experiment:2

Doc-1:	(0, 0.5773502691896257)	(1, 0.5773502691896257)	(2, 0.5773502691896257)
Doc-2:	(0, 0.5773502691896257)	(1, 0.5773502691896257)	(3, 0.5773502691896257)
Doc-3:	(2, 0.5495834326673141)	(3, 0.5495834326673141)	(4, 0.30638888950618853)
Doc-4:	(4, 0.4869354917707381)	(5, 0.8734379353188121),	
Doc-5:	(6, 0.5495834326673141)		

4.4.5 Latent Semantic Analysis results on Experiment:2

LSA which uses SVD is applied on the tfidf model. SVD preserves the highest eigen values, which are shown below:

Table 4.11: LSA for experiment:2

Doc-1:	(0, ’0.494*”beat”)	(0.428*”sweet”)	(0.383*”sugar”)
Doc-2:	(1, ’0.536*”sweet”)	(0.00*”beans”)	(-0.468*”cocoa”)
Doc-3:	(2, ’0.536*”sweet”)	(0.493*”beat”)	(0.330*”butter”)
Doc-4:	(0.780*”beat”)	(0.547*”sugar”),	(0.870*”sweet”)
Doc-5:	(4, ’0.699*”sweet”)	(0.083*”beat”)	

The five vectors of five documents are shown, the weights of all the terms are preserved. Here the weights with negative sign and zero are ignored, i.e, cocoa and beans. Thus preserving only highly weighted eigen values. The weights are removed from these terms and provided as an input to the next module.

4.4.6 Clustering results on Experiment:2

We have applied agglomerative clustering in this step. Depending upon the dataset, this time only one cluster is generated, depending upon the small size of the data set, i.e, five documents.

Table 4.12: Cluster for experiment:2

beat	sweet	sugar	butter	chocolate

4.4.7 Summary generation on Experiment:2

Upon these terms, we have applied tfidf. The sentences with highest tfidf are selected and are added to generate summary. The resultant summary is shown below:

Sweet Chocolate Butter Sugar

Hot Chocolate Cocoa Beans

4.4.8 Topic Detection on Experiment:2

The topic detection feature works with the help of LDA. Upon our summary, we have applied LDA. The topics detected by LDA are:

Table 4.13: Summary topics for experiment:2

icecream	hot	harvest	cocoa	chocolate

These topics give a generic idea to the readers about the summary.

4.4.9 Evaluation Results on Experiment-2

In this section, we have evaluated our summaries using ROUGE metric. We have compared our results with a recent topic based approach using LDA [60], while the another system is a graph based approach which uses cosine and discourse similarities to construct a graph [17]. The specification of these systems is shown in the following table.

Table 4.14: Specification of the Benchmark techniques

Specifications	Dataset	Methodology
Model-1	Novel-Dataset	Probabilistic LDA
Model-2	Novel-Dataset	Statistical and linguistic Graph based approach

4.4.10 Evaluation results on Experiment:2

We call the dataset provided by the institute as Novel-data set, which we have used on these models and our summarizer, then we have evaluated the results using ROUGE metric, which we have shown the evaluations scores in Table 4.15. In this table we have coded first Topic based approach as Model-1, second graph based approach as Model-2 and our approach as Model-3.

Table 4.15: ROUGE Scores

Models	ROUGE-1 (Recall)	Precision	FScore	ROUGE-2 (Recall)	Precision	FScore
Model-1	0.25	0.5	0.51	0.25	0.58	0.95
Model-2	0.50	0.7	0.63	0.71	0.65	0.75
Model-3	0.63	0.87	0.71	0.88	0.78	0.85

As we can see from the above results that our system shows highest ROUGE scores. The first approach shows lowest ROUGE scores as it produces summary using only one sentence. The second approach shows much higher ROUGE scores as it selects almost all sentences to form a summary which is not feasible for readers as there is no need for any summary in this case, as almost more than half of the original text is re-produced to form a summary. The third model shows highest ROUGE scores which depicts our system. Our system has selected two sentences out of five to form a summary, which is the best representative of the original data set.

Chapter 5

Conclusion and Future work

This chapter concludes the research has been carried out. A concise overview of the proposed approach in discussion section and the contribution of the approach is given. Moreover the future work is also given, which may be useful for other researchers who might be interested in pursuing or extending this work.

5.1 Conclusion

Finding out the important information that matches users interest is a problem with the growth of text-based resources, which arises the need of a system to provide them effecient and coherent summary results. A lot of research has been carried out to cater this need and variant approaches are proposed over time, i.e, graph based approaches, statistical, clustering and algebraic approaches. One of the algebraic approaches is Latent Semantic Analysis (LSA), which is used, in the proposed approach, with the combination of other techniques to produce a system for automatically generating summaries.

5.1.1 Contribution of the Research

This research solves the following problems:

1. The first and foremost issue is to create a summary that shares a common topic, to achieve this goal we have clustered semantically related terms.
2. Real world data is a combination of raw facts and figures, to remove noise, only pre-processing is not the solution. Curse of dimensionality; where different terms refer to one entity is resolved using a statistical technique called LSA. LSA uses SVD to preserve only highest eigen values, this way feature space size is reduced.
3. To give an overall idea about a summary, we have added a topic detection feature in our proposed model. This is achieved using LDA, that works on conditional probability. It iterates over the resultant summary and calculates the most contributing topic terms in the summary. This gives a user an insight about the summary and he/she gets to know what is this summary about and whether it is according to his/her interest or not.

The contribution of the research is depicted in the following table.

Table 5.1: Contribution of the proposed framework

Contribution	**Status**	**Advantages**
Preprocessing	✓	Removal of noise and raw data
Dimension Reduction	✓	Less feature size
Clustering	✓	Similar terms together
Topic detection	✓	Topic is known before reading the summary

5.1.2 Discussion

Using LSA we have reduced dimensions and to combine similar terms, we have applied agglomerative clustering algorithm. Our sentence selection algorithm is based on $tfidf$. We have integrated topic detection module to depict the most contributing topic terms in the summary, achieved by LDA. We have evaluated our system on DUC 2004 dataset on TASK 1, 2 and 4. Our results are among the best reported on this dataset for ROUGE-1, ROUGE-2 and ROUGE-SU4 metrics. Benchmarking the proposed model on DUC dataset showed that our model outperformed all competitors. We have seen distinctive results in all methodologies, previous approaches have still unresolved issues to deal issues like redundancy and finding semantic relations either between terms or sentences. We have solved these issues and explained in detail how redundancy is resolved in our approach. Previous state-of-the art techniques have poor mechanisms to deal with semantically related sentences. We resolve these core issues in our proposed approach. The semantic relations between terms is very important factor to create a comprehensive and human readable summary. If such underlying semantic relations or meanings are not identified, a summary can contain unrelated sentences which will not give readers an understandability of the summary. As one sentence contains different information while the other sentence contains different information This problem is solved in our system using clustering. Similar terms are clustered together, if more than one clusters are formed then summary is created from these clusters. Number of summary files depends upon number of clusters, as different clusters contain different information, it is not logical to create one summary of documents which share different topics. Terms in one cluster are semantically similar and are different from another clusters.

Last but not the least, in our approach, have tried to somehow tell the readers about the main contributing terms or topics in the summary, through which readers get an idea what is this summary about, is it related to their interest or not? By providing them with the topics of the summary beforehand, we save their time which is an increment in efficiency.

5.1.3 Future Work

In future, it can be an interesting direction to a): combine these approaches with other dimensions reduction measure like principal component analysis and to observe the quality of summary. b): or to further improve this approach by narrative study. The proposed work can be tested upon other data sets, e.g, 20Newsgroups. The researchers in future can work upon other clustering algorithms and combine them in the proposed approach, other clustering algorithms can be K-means or fuzzy clustering, DBscan etc.

Bibliography

[1] Ramiz M Aliguliyev. A new sentence similarity measure and sentence based extractive technique for automatic text summarization. *Expert Systems with Applications*, 36(4):7764–7772, 2009.

[2] Ramiz M Aliguliyev. Clustering techniques and discrete particle swarm optimization algorithm for multi-document summarization. *Computational Intelligence*, 26(4):420–448, 2010.

[3] Rachit Arora and Balaraman Ravindran. Latent dirichlet allocation based multi-document summarization. In *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pages 91–97. ACM, 2008.

[4] Ramakrishna Bairi, Rishabh Iyer, Ganesh Ramakrishnan, and Jeff Bilmes. Summarization of multi-document topic hierarchies using submodular mixtures. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 553–563, 2015.

[5] Elena Baralis, Luca Cagliero, Naeem Mahoto, and Alessandro Fiori. Graphsum: Discovering correlations among multiple terms for graph-based summarization. *Information Sciences*, 249:96–109, 2013.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Jaime Carbonell and Jade Goldstein. 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*, pages 335–336. ACM, 1998.

[8] Freddy YY Choi, Peter Wiemer-Hastings, and Johanna Moore. Latent semantic analysis for text segmentation. In *Proceedings of the 2001 conference on empirical methods in natural language processing*, 2001.

[9] John M Conroy and Dianne P O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407. ACM, 2001.

[10] Dipanjan Das and André FT Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[11] Dipanjan Das and André FT Martins. A survey on automatic text summarization. *Literature Survey for the Language and Statistics II course at CMU*, 4:192–195, 2007.

[12] Rajesh M Desai, Alon S Housfater, Philip E Parker, and Roger C Raphael. Domain specific representation of document text for accelerated natural language processing, February 20 2018. US Patent 9,898,447.

[13] Remco Dijkman and Anna Wilbik. Linguistic summarization of event logs–a practical approach. *Information Systems*, 67:114–125, 2017.

[14] Chris HQ Ding. A probabilistic model for latent semantic indexing. *Journal of the American Society for Information Science and Technology*, 56(6):597–608, 2005.

[15] Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *Proceedings of the Conference on Empirical*

Methods in Natural Language Processing, pages 897–906. Association for Computational Linguistics, 2008.

[16] Günes Erkan and Dragomir R Radev. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[17] Rafael Ferreira, Luciano de Souza Cabral, Frederico Freitas, Rafael Dueire Lins, Gabriel de França Silva, Steven J Simske, and Luciano Favaro. A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, 41(13):5780–5787, 2014.

[18] N Freitas and A Kaestner. Automatic text summarization using a machine learning approach. In *Brazilian Symposium on Artificial Intelligence (SBIA), Brazil*, 2005.

[19] Mahak Gambhir and Vishal Gupta. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66, 2017.

[20] Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd international conference on computational linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

[21] Yihong Gong and Xin 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*, pages 19–25. ACM, 2001.

[22] Fergyanto E Gunawan, Adrian Victor Juandi, and Benfano Soewito. An automatic text summarization using text features and singular value decomposition for popular articles in indonesia language. In *Intelligent Technology and Its Applications (ISITIA), 2015 International Seminar on*, pages 27–32. IEEE, 2015.

[23] Shengbo Guo and Scott Sanner. Probabilistic latent maximal marginal relevance. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 833–834. ACM, 2010.

[24] Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268, 2010.

[25] Andreas Hotho, Andreas Nürnberger, and Gerhard Paaß. A brief survey of text mining. In *Ldv Forum*, volume 20, pages 19–62, 2005.

[26] Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented blog summarization by sentence extraction. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 901–904. ACM, 2007.

[27] Ya-Han Hu, Yen-Liang Chen, and Hui-Ling Chou. Opinion mining from online hotel reviews–a text summarization approach. *Information Processing & Management*, 53(2):436–449, 2017.

[28] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[29] Anna Kazantseva and Stan Szpakowicz. Summarizing short stories. *Computational Linguistics*, 36(1):71–109, 2010.

[30] SSRK Kiriti, B Gogineni Siri, S Vijaya Durga Bhavani, and Ch Nanda Krishna. Automatic text summarization: Comparison of various techniques. *International Journal of Engineering Technology and Computer Research*, 5(2), 2017.

[31] Jon Kleinberg, Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. The web as a graph: measurements, models, and methods. *Computing and combinatorics*, pages 1–17, 1999.

[32] Rashmi Kurmi and Pranita Jain. Text summarization using enhanced mmr technique. In *Computer Communication and Informatics (ICCCI), 2014 International Conference on*, pages 1–5. IEEE, 2014.

[33] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Barcelona, Spain, 2004.

[34] Hans Peter Luhn. The automatic creation of literature abstracts. *IBM Journal of research and development*, 2(2):159–165, 1958.

[35] Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. Graph-based text summarization using modified textrank. In *Soft Computing in Data Analytics*, pages 137–146. Springer, 2019.

[36] Kathleen McKeown and Dragomir R Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 74–82. ACM, 1995.

[37] Mayuri Mhatre, Dakshata Phondekar, Pranali Kadam, Anushka Chawathe, and Kranti Ghag. Dimensionality reduction for sentiment analysis using pre-processing techniques. In *Computing Methodologies and Communication (ICCMC), 2017 International Conference on*, pages 16–21. IEEE, 2017.

[38] Rada Mihalcea and Hakan Ceylan. Explorations in automatic book summarization. In *EMNLP-CoNLL*, volume 7, pages 380–389, 2007.

[39] Rada Mihalcea and Paul Tarau. Textrank: Bringing order into text. In *EMNLP*, volume 4, pages 404–411, 2004.

[40] Mohamed Atef Mosa, Arshad Syed Anwar, and Alaa Hamouda. A survey of multiple types of text summarization with their satellite. 2018.

[41] Fionn Murtagh and Pierre Legendre. Wards hierarchical agglomerative clustering method: which algorithms implement wards criterion? *Journal of classification*, 31(3):274–295, 2014.

[42] Joel Neto, Alex Freitas, and Celso Kaestner. Automatic text summarization using a machine learning approach. *Advances in Artificial Intelligence*, pages 205–215, 2002.

[43] Joel Larocca Neto, Alex A Freitas, and Celso AA Kaestner. Automatic text summarization using a machine learning approach. In *Brazilian Symposium on Artificial Intelligence*, pages 205–215. Springer, 2002.

[44] Joel Nothman, Hanmin Qin, and Roman Yurchak. Stop word lists in free open-source software packages. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 7–12, 2018.

[45] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab, 1999.

[46] Scott M Petry, Ramesh Rajagopal, Peter K Lund, Fredric L Cox, Adam P Moore, Leslie L Dunston, Varley H Taylor, Zachary L Segal, Luka I Stolyarov, Joshua R McMains, et al. Secure analysis application for accessing web resources via url forwarding, July 17 2018. US Patent App. 10/027,700.

[47] Anjusha Pimpalshende and AR Mahajan. Test model for stop word removal of devnagari text documents based on finite automata. In *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*, pages 672–674. IEEE, 2017.

[48] Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.

[49] Nazreena Rahman and Bhogeswar Borah. A spell correction method for query-based text summarization. In *Proceedings of the International*

Conference on Computing and Communication Systems, pages 337–345. Springer, 2018.

[50] Radim Rehurek and Petr Sojka. Software framework for topic modelling with large corpora. In *In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Citeseer, 2010.

[51] Wee Meng Soon, Hwee Tou Ng, and Daniel Chung Yong Lim. A machine learning approach to coreference resolution of noun phrases. *Computational linguistics*, 27(4):521–544, 2001.

[52] Josef Steinberger and Karel Ježek. Evaluation measures for text summarization. *Computing and Informatics*, 28(2):251–275, 2012.

[53] Josef Steinberger, Mijail A Kabadjov, Massimo Poesio, and Olivia Sanchez-Graillet. Improving lsa-based summarization with anaphora resolution. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, 2005.

[54] Josef Steinberger and M Křišt'an. Lsa-based multi-document summarization. In *Proceedings of 8th International PhD Workshop on Systems and Control*, volume 7, 2007.

[55] Shiliang Sun, Chen Luo, and Junyu Chen. A review of natural language processing techniques for opinion mining systems. *Information Fusion*, 36:10–25, 2017.

[56] Kristina Toutanova, Chris Brockett, Michael Gamon, Jagadeesh Jagarlamudi, Hisami Suzuki, and Lucy Vanderwende. The pythy summarization system: Microsoft research at duc 2007. In *Proc. of DUC*, volume 2007, 2007.

[57] Giang Tran, Eelco Herder, and Katja Markert. Joint graphical models for date selection in timeline summarization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics,*

volume 1, pages 1598–1607. Association for Computational Linguistics, 2015.

[58] Xiaojun Wan and Jianwu Yang. Multi-document summarization using cluster-based link analysis. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306. ACM, 2008.

[59] Sicui Wang, Weijiang Li, Feng Wang, and Hui Deng. A survey on automatic summarization. In *Information Technology and Applications (IFITA), 2010 International Forum on*, volume 1, pages 193–196. IEEE, 2010.

[60] Zongda Wu, Li Lei, Guiling Li, Hui Huang, Chengren Zheng, Enhong Chen, and Guandong Xu. A topic modeling based approach to novel document automatic summarization. *Expert Systems with Applications*, 84:12–23, 2017.

[61] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[62] Chandra Yadav and Aditi Sharan. A new lsa and entropy-based approach for automatic text document summarization. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 14(4):1–32, 2018.

[63] Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50, 2014.

[64] Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50, 2014.

[65] Libin Yang, Xiaoyan Cai, Yang Zhang, and Peng Shi. Enhancing sentence-level clustering with ranking-based clustering framework for theme-based summarization. *Information sciences*, 260:37–50, 2014.

[66] Laura A Yarbro and Stanley N Deming. Selection and preprocessing of factors for simplex optimization. *Analytica Chimica Acta*, 73(2):391–398, 1974.

[67] Jaya Kumar Yogan, Ong Sing Goh, Basiron Halizah, Hea Choon Ngo, and C Puspalata. A review on automatic text summarization approaches. *Journal Of Computer Science*, 12(4):178–190, 2016.

[68] Markus Zopf, Eneldo Loza Mencía, and Johannes Fürnkranz. Which scores to predict in sentence regression for text summarization? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, volume 1, pages 1782–1791, 2018.