ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

AUTOMATIC AMHARIC TEXT SUMMARIZATION USING LATENT SEMANTIC ANALYSIS

By: Melese Tamiru

A THESIS SUBMITTED TO

THE SCHOOL OF GRADUATE STUDIES OF THE ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT FOR THE DEGREE OF MASTERS OF SCIENCE IN COMPUTER SCIENCE

October, 2009

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULITY OF INFORMATICS
DEPARTMENT OF COMPUTER SCIENCE

AUTOMATIC AMHARIC TEXT SUMMARIZATION USING LATENT SEMANTIC ANALYSIS

By: Melese Tamiru

ADVISOR:

Mulugeta Libsie (PhD)

APPROVED BY

EXAMINING BOARD:

1. Dr. Mulugeta Libsie, Advisor _____
2. _____ _____
3. _____ _____

There is life beyond perfection!

Acknowledgements

First and foremost, Glory to God. Without God‟s support and guidance in my life, nothing would have been possible. I am also very grateful to my advisor, Dr. Mulugeta Libsie. He is not only my advisor but he is also my mentor. Becoming an academician has always been my dream since my childhood and this dream got refreshed when I happened to know a person named Dr. Mulugeta Libsie. Of all the qualities of Dr. Mulugeta, his kindness touched me most. Thanks Dr. Mulugeta for being who you are.

I know I tend to be a nuisance when things are not going in my way and putting up with this is very difficult. Fortunately, I am blessed with a wonderful family who has a limitless tolerance to all my agitations. I really appreciate this with lots of love and respect. I also owe a great deal to my beloved sister, Amarech Tamiru and her husband for their material and financial support.

My thanks also go to Eskinder Mekonnen, Abraham Redai, Fitsum Yitbarek, Getachew Assefa, Tesfaye Tamiru, and Fitsum Seyoum who assisted with the annotation of the data set used for the evaluation of the summarizer. I would also love to thank Tessema Mindaye who let me use the data preprocessing tools that he developed. Thanks also to my instructors and collegues who fostered an atmosphere of experimentation and exploration during the course of this study. Finally, my deepest gratitude goes to the all time friend of mine, Kalkidan Yigezu. I just can‟t thank you enough. May God bless you.

Table of Contents

List of Tables

List of Figures

List of Algorithms

List of Acronyms

ANSI	American National Standards Institute
AW	Augmented Weight
BW	Binary Weight
DUC	Document Understanding Conference
EF	Entropy Frequency
FW	Frequency Weight
GFIDF	Global Frequency Inverse Document Frequency
HITS	Hyperlink-Induced Topic Search
IDF	Inverse Document Frequency
IR	Information Retrieval
LSA	Latent Semantic Analysis
LSI	Latent Semantic Indexing
LW	Logarithm Weight
NMF	Non-negative Matrix Factorization
PLSI	Probabilistic Latent Semantic Indexing
RST	Rhetorical Structure Theory
SDD	Semi-Discrete matrix Decomposition
SVD	Singular Value Decomposition
ROUGE	Recall-Oriented Understudy for Gisting
VSM	Vector Space Model

Abstract

With the continuous increase in the number of electronic documents, the need for faster techniques to assess the relevance of documents emerges. An ideal summary is one that conveys to the reader the main themes of the document and consequently the reader can determine whether the complete document is of any relevance. Automatic Text Summarization is a technique where a program summarizes a text. A text is given to the program and the program returns a shorter and less redundant extract of the original text.

In this thesis, two generic text summarization methods that create text summaries by ranking and extracting sentences from the original documents are proposed. The first method, TopicLSA, employs Latent Semantic Analysis (LSA) to identify the main topics of a document. The identified topics along with document genre information are used to select semantically important sentences for summary generation. The second method, LSAGraph, combines Latent Semantic Analysis with graph-based ranking algorithms to compute the relevance of sentences for summary inclusion. Moreover, LSAGraph uses document genre information to penalize sentences that do not belong to the main topic of the document.

In order to evaluate the performance of the proposed summarization methods, a prototype Amharic news text summarization system is built based on the proposed methods. Evaluation of the summarization system is then conducted by comparing the system‟s summaries with manual summaries that are generated by six independent human evaluators. Despite the very different approaches taken by the proposed methods to generate a summary, both produced quite comparable performance scores.

To have an idea of the relative success of the proposed summarization methods, evaluation of the summarization system also included comparison of the proposed methods with previous summarization methods based on LSA and graph-based ranking algorithms. The results of the evaluation have shown that the proposed summarization methods have performed significantly better than previous summarization methods based on LSA and graph-based ranking algorithms.

Keywords: Summarization, Latent Semantic Analysis, Graph-based ranking algorithms.

CHAPTER ONE: INTRODUCTION

1.1 Background

The rapid growth of broadcast systems, the Internet and online information services has made the possibility of accessing enormous amounts of information very easy. In fact, the volume of literature published annually in a specific field is generally far too large for an individual to read and assimilate. Thus, the increasing availability of information has created the necessity of new technologies that can help the reader to go through vast volumes of information in a very short time.

The technology of automatic text summarization is one tool that can help users manage the vast quantity of information available. Ideally, a summary would convey to the reader the main themes of the document and consequently the reader can determine whether the complete document is of any relevance. In short, reading the summary of a document could suffice to sufficiently inform the reader about the document, or instead, could indicate to the reader that the particular document is not of interest. Text summarization can be conducted through human professionals but this is time consuming and very costly. Hence, this calls for automating text summarization.

Automatic text summarization is the task of producing the most important content from a given text document to the user in a condensed and human-readable form [2]. It makes use of a computer to extract the most relevant parts of a document and create a summary that is shorter than the original document. Hence, it avoids the need of human professionals and reduces the cost needed to produce a summary.

Usually, summaries produced by automatic text summarization are of two types: query oriented or generic. A query oriented summary presents the contents of the document that are closely related to the initial search query and it is often achieved by extending conventional Information Retrieval (IR) technologies. On the other hand, a generic summary provides an overall sense of the document‟s contents. As neither query nor topic will be provided to the summarization process, it is challenging to develop a high quality generic summarization method and even more challenging is to objectively evaluate the method [25].

Furthermore, automatic text summarization can be grouped into two categories: extraction and abstraction. Summarization by extraction produces a summary which consists of sentences extracted from the document while summarization by abstraction produces a summary which may employ words and phrases that may not appear in the original document. Extractive summarization is simpler than abstractive summarization and currently it is the general practice among researchers in the area of automatic text summarization [7]. In light of this, the summarization system that we aim to implement in this thesis is generic extractive summarization system.

There are various areas where automatic text summarization can be of great use. For instance, automatic text summarization can be used to prepare information for use in small mobile devices, such as a PDA, which may need considerable reduction of content. Also, search engines could index summaries instead of the whole document, lowering the resources needed by indexing algorithms. This also improves the performance of the search engine in terms of correctly responding to user queries [1].

1.2 Statement of the Problem

The continuing growth of the World Wide Web and on-line text collections make a large volume of information available to users. News is one such information that is highly characterized as an international information requirement and hence service providers in this case are numerous [33]. This being the case, access to relevant news items has become a significant problem.

A growing number of Amharic news service providers are publishing their content online. To mention a few, the Ethiopian Reporter, Addis Admas, and Addis Zena have been updating their websites regularly with news of all kinds in an average of 15-20 articles per week. Given the rate of production and the sheer volume of online newspapers, presenting the user with a summary of each newspaper greatly facilitates the task of finding the desired newspapers.

Though the field of automatic text summarization has enjoyed a lot of research for many languages, Amharic language and in general local languages are underrepresented in the area. However, few researchers have attempted to develop automatic summarization system for Amharic [33, 34, 36, 55, 56, 59]. So far all these summarization works for Amharic texts are based on pure statistical methods and machine learning algorithms. However, pure statistical methods fail to capture the

main topics of a document as they are ignorant of the semantics of the words in the document. Thus, sentences that reflect the main topic of the document might not be selected in the summary.

Machine learning algorithms, on the other hand, require a great deal of training corpora and hence, they are very costly. Moreover, summarization systems based on machine learning algorithms are not easily adaptable to other languages or domains. In this thesis we attempt to consider the contextual meaning of words in order to capture the central topics of a document and thus, summaries of high quality can be produced. This is achieved through the use of Latent Semantic Analysis (LSA) and graph-based ranking algorithms. Furthermore, these methods do not need any training corpora which make them easily adaptable to different languages and genre.

LSA is a technique used to obtain the semantic representation of terms, sentences, or documents on the basis of their contextual use. The method relies on the fact that there is a substantial amount of semantic content associated with most text strings that is not explicit in those strings or in the mere statistical co-occurrence of the strings with other strings, but which is nevertheless extremely relevant to the text [13].

Graph-based ranking algorithms, on the other hand, represent a document using a graph in which a vertex is added for each sentence in the document and edges between the vertices are established using sentence similarities [47, 48]. Thus, both methods are believed to assist in the identification of semantically similar terms and sentences which entail the main topic of the document. The main aim of this thesis is thus, to find out how these methods can be used for summarizing Amharic news text.

1.3 Objectives

The general and specific objectives of this thesis are described below.

1.3.1 General Objective

The general objective of this thesis is to investigate the application of Latent Semantic Analysis for automatic summarization of Amharic news texts.

1.3.2 Specific Objectives

To achieve the general objective of the study, the following specific objectives are identified.

- To review literature and analyze state of the art methods for text summarization and identify their pros and cons.
- To review literature on Latent Semantic Analysis and graph-based ranking algorithms in the context of information retrieval and text summarization, and identify their pros and cons.
- To design a generic model for LSA-based automatic Amharic text summarizer.
- To develop an algorithm that extracts semantically important sentences.
- To develop a prototype Amharic text summarizer.
- To conduct experiments to evaluate the usability of the proposed system.

1.4 Scope and Limitations

This thesis focuses on developing generic extractive summarization for Amharic news texts using LSA. It doesn‟t employ language dictionaries and deep linguistic analysis is not considered. This thesis further focuses on the particular nature of news texts to enhance the use of LSA for summarization. That being the case, some results of the thesis might not be applicable for summarization of other document genres. Furthermore, this thesis considers only Amharic textual documents that contain sequence of Amharic alphabets without any figure, table, image or pictorial representations.

1.5 Methodology

In order to achieve the objective of the study, the following methods are employed.

1.5.1 Literature Review

Extensive literature review is conducted to get a deeper understanding of automatic text summarization with specific emphasis on the use of Latent Semantic Analysis and graph-based ranking algorithms for text summarization and information retrieval.

1.5.2 Data Corpus

The dataset used for evaluating the proposed summarization system contains 50 Amharic news items whose lengths are in the range of 17 to 44 sentences. These news items were collected from the Web site of the Amharic version of the Ethiopian Reporter. Short news items with less than 17 sentences were not used in the evaluation due to the fact that summarizing short articles does not make much sense in real applications. Though evaluating the system with news stories containing more than 44 sentences is very desirable, it was very difficult to obtain such news stories. Furthermore, the news articles used for the evaluation are on different domains (politics, sport, technology) which helped to evaluate the performance of the system for different domains.

Six independent human evaluators were employed to conduct manual summarization on the 50 documents contained in the data corpus. For each document, two summaries at 20% and 30% compression rate were prepared.

1.5.3 Development Tools

In order to accomplish the study, different tools are employed. Java programming language is used for the development of the prototype. Java is selected for its suitability in developing standalone applications and because it supports Unicode encoding. We have used JAMA, a free Java library package, for constructing matrices and for all computations associated with matrix.

1.5.4 Evaluation

Performance evaluation was conducted by comparing the system summaries with their corresponding manual summaries. We have also evaluated the performance of our summarization system in relative to previous LSA-based summarizers and graph-based summarizers which we developed for this purpose. The evaluation metrics used are the well known IR metrics, precision, recall, and F-Score.

1.6 Expected Contribution

The main contribution of this thesis focuses on finding an efficient method of automatic summarization system for Amharic news texts. This study, apart from being an academic exercise,

is believed to have produced results that have indicated the possible application of Latent Semantic Analysis for automatic summarization of Amharic news texts. Hence, this research is assumed to contribute a significant value to the development of a full-fledged automatic summarizer for Amharic documents.

Furthermore, the application of LSA for Amharic text summarization is expected to pave the way for using the theory of LSA in applications based on Amharic language processing. This includes information filtering from Amharic documents, classification of Amharic documents, cross-language IR in which Amharic language is one component, and automatic evaluation of Amharic essays.

1.7 Thesis Outline

The rest of this thesis is organized as follows. In Chapter 2, background information of automatic text summarization is described. The Chapter explains different types of summarization and various approaches to automatic text summarization. Theoretical background of Latent Semantic Analysis and graph-based ranking algorithms is also presented in this Chapter. Chapter 3 critically reviews related summarization works based on LSA and graph-based ranking algorithms. The Chapter also proposes a solution to bridge the gaps identified in related works and previous summarization works for Amharic language.

Chapter 4 presents a detailed description of our proposed approaches to summarization along with a prototype LSA-based Amharic summarizer. Chapter 5 presents the empirical results of the proposed system along with their interpretations. Finally, Chapter 6 concludes the thesis with the research findings, conclusions, and future works.

CHAPTER TWO: LITERATURE REVIEW

In this Chapter, we provide a brief overview of the field of automatic summarization. There are several types of summarization and some of them are explained in this Chapter. This Chapter also describes the most important stages of automatic text summarization. Furthermore, we investigate state of the art techniques used in the area of text summarization. Automatic text summarization is an active field of research and hence, there are a lot of works in the area. Here, we will only present those works whose contribution made a great progress to the field.

We believe that background information on the theory of Latent Semantic Analysis and graph-based ranking algorithms is necessary as our proposed summarization system is based on them. Thus, this Chapter explains the model of Latent Semantic Analysis and graph-based ranking algorithms from information retrieval and summarization perspectives. Evaluating the performance of a summarization system is a very challenging task and over the years various methods have been proposed. Principal methods of evaluating summaries are also reviewed in this Chapter.

2.1 Basic Notion of Automatic Text Summarization

Automatic text summarization is a process by which the most important concepts in a document are identified and then presented in a condensed and human-readable form. Its main objective is to reduce the complexity and length of the original text while retaining the main ideas of the text [3]. The produced summary should be non-repetitive and should give as much precise information as possible. In short, the summary should allow the reader to answer questions about the main topics in the given text or work as a reference pointer to parts of the original text [1, 2].

Over the years, various researchers have defined a summary in different ways and the most notable one is that defined by Hovy [4]. According to Hovy, a summary is a text that is produced from one or more texts, that contains a significant portion of the information in the original text(s), and that is no longer half of the original text(s). Based on the process of generating the summary rather than what it should contain, Jones and Karen [5] defined a summary as a reductive transformation of a source text into a summary text by extraction or generation.

Still another definition is that of Luhn [3] who is a pioneer researcher in the area. Luhn defined text summarization as the process of distilling the most important information from a source text to produce an abridged version for a particular user/users and task/tasks.

Automatic text summarization has been around for more than 40 years now. As the amount of information on the Internet is growing abundantly, the need for text summarization is even greater today. People have access to vast amount of information and yet not enough time to digest all of it. Thus, summaries can reduce the time needed to absorb the key facts in a document. The need for text summaries has been clearly described in the words of the American National Standards Institute (ANSI) – "A well prepared abstract enables readers to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether they need to read the document in its entirety" [6].

The application areas of automatic text summarization are extensive. Automatic summaries could be displayed in search results as an information tool for the user. Users of digital libraries and journals could benefit from summaries as they can find relevant text easily. News portals could provide precise summaries about news that emerged from multiple source articles. Web browsing could be better if summaries of web pages are available. Moreover, other than being used as tools for users, search engines could index summaries instead of the whole document which improves the performance of search engines in terms of relevance [1].

2.1.1 Types of Summaries

Many types of summaries have been identified which in general fall to either indicative or informative categories. An indicative summary is one that provides an idea of what the document is about or it indicates the document‟s relevance to the reader. That is, it gives abbreviated information on the main topics of a document by preserving the most important passages of the document. Indicative summaries are often returned by search engines as a response to user queries. Hence, they are only meant to help the user decide whether or not to read the full document. On the other hand, the purpose of informative summaries is to deliver as much information as possible to the user and to serve as a substitute for the full document. The typical lengths of indicative summaries range between 5 to 10% of the full document and that of informative summaries range between 20 to 30% of the complete text [7].

Furthermore, the field of summarization has witnessed two general approaches of summarization: abstraction and extraction. In abstraction, summaries are created by generating words and phrases which may not be present in the original document and such summaries are called abstracts. Extraction focuses on reusing portions of the document such as words, sentences or paragraphs to make a summary or an extract. The vast majority of current summarization systems perform an extractive summarization. This is because abstraction relies on a deep understanding of natural languages which is a very challenging task that is not successfully achieved yet [7].

It is also possible to classify summaries into two categories: of single document and multiple documents. The difference between the two is mainly in the input document. In case of single document summarization, a single document is condensed to form an abridged version of it. On the other hand, multi-document summarization deals with summarizing a collection of thematically related documents. Summarizing a single document is a challenging task but this is even more with multi-document summarization. This is attributed to the fact that in case of summarizing multiple documents, repetitions and inconsistencies between documents have to be well accounted for. Due to this reason, multi-document summarization is much less developed than single document summarization [4].

Another criterion to classify summaries is based on their purpose. That is, a summary can be generic which tries to represent all relevant features of a source text or it can be query-driven reflecting on only some topics in the input text that are specific to a given query. In short, generic summaries are text-driven whereas query-driven or user-focused ones rely on the specification of the user‟s information need, like a question or keywords [25]. Until recently, generic summaries were more popular, but with the prevalence of full-text searching and personalized information filtering, user-focused summaries are gaining importance [30]. In this thesis, our aim is to produce generic summaries which are informative enough to be used as a substitute to a single document.

2.1.2 The Stages of Automatic Text Summarization

According to Hovy [4] there are three distinct stages in performing text summarization: topic identification, interpretation, and generation. However, most systems today use the first stage only. The first stage, topic identification, is a process that determines the most important units such as words, sentences or paragraphs from a given document. Topic identification usually starts with

document pre-processing where the document undergoes several processes to be represented by terms capable of representing the content of the document. It, then, applies several summarization techniques ranging from simple statistical methods to complex methods that employ natural language understanding to select important units of a document.

Almost all systems employ several independent modules to perform the first stage. After each module assigns a score to each unit of input, a combination module combines the scores for each unit and assigns a single integrated score to it. The system, then, returns the highest-scoring units based on the required length of the summary.

The second stage, interpretation, is performed only by summarization systems which are based on abstraction. During this stage, the important topics identified in the first stage are fused, represented in new terms, and expressed using new concepts or words that may not be found in the original text. To accomplish this, additional knowledge about the task and audience of the summary along with prior knowledge about the subject domain is required. But due to the difficulty of building enough domain knowledge, very few summarizers to date have performed interpretation.

The generation phase is mainly performed to improve the readability of the final summary output. In the case of summarization through abstraction, the results of the interpretation phase are usually abstract representations that are unreadable to humans. Thus, summarization systems require the techniques of natural language generation to produce a human-readable text. On the other hand, extract summaries suffer from coherency due to dangling references, omitted text linkages, and repeated or omitted textual units. Hence, in the case of extract summaries, there is no need of generation stage but a process of smoothing can be used to make the extracted pieces coherent. The process of smoothing identifies and repairs possible causes of text incoherency [4, 8].

In the sequel, we will describe basic operations that are performed in the topic identification stage of summarization considering its relevance to our thesis. Concepts related to automatic representation of documents and queries such as term extraction, sentence extraction, and term weighting will be briefly discussed.

The most crucial operation required in information retrieval or in natural language processing at large is assigning appropriate terms and identifiers capable of representing the content of a

collection of documents. This task, which is known as indexing, can be performed manually by trained experts or it can be performed automatically in modern environments. The process of automatic indexing is composed of two major tasks. The first task is extracting terms or concepts from each document which are capable of representing the content of the document. The second task is assigning each term a weight or value that signifies its importance for the purpose of content description [39].

A. Index Term Extraction

The task of index term extraction follows a series of activities. Each of them is explained below.

Lexical Analysis

Lexical analysis starts with tokenization which is the identification of all the individual words that constitute the input text. That is, given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. Tokenization can occur at a number of different levels: a text could be broken up into paragraphs, sentences, words, syllables, or phonemes. Punctuation marks and spaces are usually used to infer the beginning and the end of a token. For instance, the procedure for identifying words in Amharic documents makes use of the Amharic word separators such as single space, netela serez (፣), hulet neteb (:), dereb serez (፤), arat neteb (፨), carriage return, line feed , tab, etc.

Lexical analysis also incorporates a sort of text cleaning process in addition to tokenization. The text cleaning process removes numbers and symbols such as 2000, $, @, %, #, etc. which do not make up good index terms. It also converts abbreviations and acronyms into their full text, and merges hyphenated words. For instance, the hyphenated word **ክፍለ-ከተማ** will be treated as a single word **ክፍለከተማ** and the abbreviation **ዓ.ም** is expanded to **ዓመተ ምህረት**. The text cleaning process helps to avoid errors which are against the syntactical rules of the language under consideration [39, 40, 41].

Normalization

There are different Amharic letters that have the same sound and such letters might be used interchangeably in a given word. For instance, the words **ስራ** and **ሥራ** both mean “work” which

shows how the characters **ሰ** and **ሥ** can be used interchangeably. Normalization handles such type of inconsistency in writing words by changing characters of the same sound to a common form. This avoids the unnecessary representation of a given word in different forms [39, 40, 41]. This is especially useful in keyword identification which is employed in this thesis to identify important sentences in a given document.

Stop-Word removal

The words of a document text do not have equal value for indexing purpose. Some words are lexical devices that serve grammatical purposes and do not refer to objects or concepts. The common words in English such as of, a, and the, are stop-words and such words are generally used to "glue" sentences together but they usually do not carry meanings. Thus, such words could be removed from the text by comparing each term in the text with a list of common words developed for a particular language and sometimes for a particular domain.

Amharic language has also its own stop-words such as **ሆነ**, **ሁሉ**, **ነው**, **ነበር**, etc. There are also news specific words such as **አስታውቀዋል**, **ይካሄዳል**, **ተጠናቀቀ**, etc. which can be treated as stop-words. From summarization point of view, removing such words from the documents guarantees that sentences will not be favored for inclusion in a summary just because they contain highly occurring stop-words. Stop-word removal also reduces the complexity of the document representation and the number of tokens to be processed [39, 40, 41].

Stemming

After removing the stop-words in a document, the remaining words are stemmed to their root form if they have morphological variants. This is based on the assumption that words with the same stem are semantically related and have the same meaning to the user of the text**.** Furthermore, bringing varieties of a word to a common form reduces the number of different terms needed for representing a document which saves storage space and processing time. For example, the following six variants **ቤቱ**, **ቤቶች**, **ቤታችን**, **ቤቶቻችን**, **ቤታቸው**, **ቤቶቻቸው** are changed into their stem word **ቤት** [39].

Various attempts have been made to develop a stemming algorithm for the Amharic language [41, 42, 43] and in this thesis we will make use of the stemmer developed in [41] as it is the only stemmer whose source code is readily available.

B. Term Weighting

After performing tokenization, normalization, stop-word removal, and stemming, the next step is to find the weight of the terms according to their importance in representing a document. Not all terms are equally important in reflecting the content of a specific text and thus, an importance indicator or a term weight should be associated with each index term. There are many weighting functions and most of them rely upon the distribution pattern of the terms within a document as well as in the document collection as a whole. The weighting functions use these distribution statistics to compute the local (within a document) and the global (within a document collection) weight of each term. The weight of a term is then found by taking the product of the term"s local weight and global weight.

When term weighting is applied in text summarization, the local weight of a term reflects the importance of the term in the sentence containing the term and the global weight reflects the term"s importance in the document. More specifically, the weight of term j in sentence i, a_{ij}, is calculated as follows [13]:

$$a_{ij} = L(t_{ij}) \cdot G(t_{ij}) \tag{2.1}$$

where, t_{ij} denotes the frequency with which term j occurs in sentence I,

$L(t_{ij})$ is the local weight for term j in sentence I, and

$G(t_{ij})$ is the global weight for term j in the whole document.

Major local and global weighting functions that are used in information retrieval are described below [23, 13].

Local Weighting

Local weighting has the following four alternatives:

- Frequency Weight (FW): $L(tij) = tf_{ij}$, where tf_{ij} is the number of times term j occurs in sentence i.
- Binary Weight (BW): $L(t_{ij}) = 1$, if term j appears at least once in sentence i; $L(t_{ij}) = 0$, otherwise.
- Augmented Weight (AW): $L(t_{ij}) = 0.5+0.5(tf_{ij}/tfmax_i)$, where $tfmax_i$ is the frequency of the most frequently occurring term in the sentence.
- Logarithm Weight (LW): $L(t_{ij}) = \log(1+ tf_{ij})$.

The most common local weighting function is frequency weight. It is based on the assumption that the importance of a content term (after stop-word removal) in describing the topic of a document is determined by the frequency of the term in the document. That is a content term that appears more frequently in a text is more important than a rarely appearing term. Raw frequency weight does not give any distinction between the occurrences of a rare term in a short sentence (document, in the context of Information Retrieval) and in a long sentence. However, the occurrence of a rare term in a short sentence is more significant than its occurrence in a long sentence. Hence, the logarithm, binary and augmented weight are often used to smooth this bias [23].

Global Weighting

Global weighting has the following four possible alternatives:

- Inverse Document Frequency (IDF): $G(t_{ij}) = \log (N/n_j) +1$, where N is the total number of sentences in the document, and n_j is the number of sentences that contain term j.
- Global Frequency Inverse Document Frequency (GFIDF): $G(t_{ij}) = gf_j/sf_j$, where the sentence frequency sf_j is the number of sentences in which term j occurs, and the global frequency gf_j is the total number of times that term j occurs in the whole document.
- Entropy Frequency (EF): $G(t_{ij}) = 1 - \sum_i \frac{pij \log(pij)}{\log(nsent)}$, where $p_{ij} = tf_{ij}/gf_j$ and nsent is the number of sentences in the document.

All of the global weighting functions basically give less weight to terms that occur frequently or in many sentences. IDF and GFIDF are closely related, both assign a high degree of importance to terms occurring in only a few sentences of a document. However, GFIDF increases the weight of

frequently occurring terms. In addition, neither weighting function considers the distribution of terms over sentences. EF makes use of information theory to measure the importance of a term. It assigns minimum weight to terms that are equally distributed over sentences and maximum weight to terms which are concentrated in a few sentences. EF, unlike IDF and GFIDF, takes into account the distribution of terms over sentences [61].

In this thesis, we will investigate the influence of all combinations of these local and global weights on the performance of our summarization system.

2.1.3 Approaches to Text Summarization

Since the introduction of the field of automatic text summarization by Luhn [3], various approaches of summarization have been proposed and most are based on sentence extraction or selection. Different ways of classifying these approaches can be found in the literature and here, we will make use of the classification method presented in [6] to offer a brief overview of the traditional techniques used in automatic text summarization. This classification is based on the level of processing required to build a summary. Accordingly, three categories are identified as surface, entity, and discourse levels.

Surface Level Approaches

The earliest works in automatic summarization used surface level approaches to decide which parts of a text are important. The oldest known automatic summarization is that of Luhn [3], wherein term frequencies were used to measure sentence relevance. The basic assumption here is that the most frequent words in a text are the most representative of its content, and hence fragments of text containing them are more relevant. However, not all words in a text are important and to this end, words beyond high and low frequency as well as those words contained in a stop word list are left out of consideration. The remaining words in the document are then sorted alphabetically so that pairs of succeeding words can be compared letter by letter. This allows for similar words to be found (e.g., differ, difference, different). These similar words thereby attained are called significant words.

Next, the significance factor of a sentence is computed using the formula:

$$sf = \frac{(the\ number\ of\ significant\ words)^2}{the\ total\ number\ of\ words} \tag{2.2}$$

Then, sentences with the highest significance factor are extracted to produce a summary or what Luhn calls as "auto-abstracts". The results obtained by Luhn were neither good condensations nor very coherent texts though he believed his "auto-abstracts" were satisfactory indicative abstracts for papers within the science and technology fields [10].

The work of Luhn was further extended to include three new parameters for calculating the weights of sentences. These were sentence position in a text, cue phrases, and title and heading words. The position of a sentence in a text, in general, is believed to reflect the importance of the sentence. For example, newspaper articles have the most important sentences at the beginning of the article while technical documents have the most important sentences in the conclusion section. In fact, a very simple and surprisingly successful method for summarization is the selection of the first sentences in a text. Various researchers have reported that this simple method of taking the lead (first sentence in the first paragraph) as summary often outperforms other methods, especially with newspaper articles [2, 11].

Cue phrases are words or phrases that signal whether a sentence is important or not. In general there are three categories of cue phrases: bonus, stigma, and null phrases. Bonus phrases are used to emphasize the importance of a sentence in a text while stigma phrases reflect that the sentence is not important. Null phrases are neutral phrases and are not considered when the weight of a sentence is computed. Few examples of bonus phrases are „significantly", „in conclusion", „in this paper we show", etc. whereas „hardly" and „impossible" are examples of stigma phrases. Thus, each cue phrase is assigned a positive or negative relevance. The weight of each sentence is then the sum of the weights of the words in it.

Sentences can also be scored for containing words that appear in the text"s title or headings, or in the user"s query, for a query based summary. The basic idea behind this assumption is that authors usually use informative titles which could reveal the subject matter of the document. Using the combination of the features cue-words, title words, and the position of a sentence to generate summaries were shown to produce successful results in various studies [11, 33, 34, 36].

Surface level approaches can also be tailored to a specific domain or corpus to give corpus based approach for text summarization. Corpus based approaches try to determine the relevance of words by first assigning the document to a particular domain. This helps to determine certain common terms in a given field that do not carry salient information and hence, their relevance can be reduced. It was further proved that the relevance of a term in a document is inversely proportional to the number of documents in the corpus containing the term. The formula to compute term relevance is given by (tf_i x idf_i) where tf_i is the frequency of term i in the document and idf_i is the inverted frequency of documents containing this term. Sentences can then be scored by the sum of term relevance in the sentence [7, 12].

Term relevance can also be measured by counting concepts rather than mere term counting. By making use of an electronic thesaurus or WordNet, each word in the text is associated to a more general concept, and frequency is computed on concepts instead of particular words. For instance, the occurrence of the concept "bicycle" is counted when any of the words "bicycle", "bike", "pedal", or "brake" is found [13, 14].

Furthermore, surface level approaches in combination with machine learning algorithms have resulted in more advanced summarization systems. Particularly, such systems use a Bayesian classifier algorithm to compute the probability of a sentence in a source document being included in a summary. The classifier is first trained using a corpus of several pairs of full documents/summaries. These summaries are abstracts created by professional abstractors. The Bayesian formula uses different statistical features to compute the probability of a sentence being relevant. Statistical features that the Bayesian formula uses are usually sentence length, cue phrases, position of a sentence in a paragraph, most frequent words (thematic words) and proper names [15, 33].

Entity Level Approaches

Entity level approaches model text entities and their relationships to capture patterns of connectivity in a text which may be used to determine salient information. In general, words can be connected in various ways, including repetition, co-reference, synonymy, and semantic association as expressed in thesauri. The degree of connectedness of words can then be used to score sentences and paragraphs. In essence, more connected sentences are assumed to be more important. With this

approach, it has been shown that the main drawback of surface level approaches, which is lack of coherence and cohesion, can be resolved [7].

Various automatic summarization methods employ text connectivity to summarize a document and of these, lexical chain which is first introduced by Barzilay and Elhadad [17], is to be mentioned. Summarization based on lexical chains first selects a set of candidate words and then, for each candidate word an appropriate chain is computed. Cohesive relation (i.e., repetition, synonymy, antonymy, hypernymy, and holonymy) between terms is a criterion for chain formation. A lexical chain, therefore, is a chain of words in a text such that each word in the chain bears some kind of cohesive relationship to a word that is already in the chain [2].

Once the source text is represented using lexical chains, the strength of each lexical chain is determined on the basis of the number and type of relations in the chain. A summary is then built by selecting sentences where the strongest chains are highly concentrated. This is in an assumption that picking sentences represented by strong lexical chains gives a better indication of the central topic of a text than simply picking the most frequent words in the text.

Another way of using text connectivity for summarization is based on phrasal analysis and anaphoric relations in text. Here, the main aim is to identify those phrasal units across the entire span of the document that best function as representative highlights of the document‟s content. One way to achieve this is by using co-reference resolution system. Co-reference resolution is the process of determining if two expressions in natural language refer to the same entity [7]. Once the desired phrasal units are identified, they are combined to form “capsule overviews”. The capsule overview is not a sequence of sentences as is expected in a summary but it is a list of key-phrases preserving the flow narration in the original document [19].

Discourse Level Approaches

Before we delve into summarization approaches based on discourse structure, let‟s first define what a discourse is. Discourse is understood to refer to any form of language-based communication involving multiple sentences or utterances such as text and dialogue [20]. Discourse level approaches exploit the discursive organization of a text to improve the relevance and quality of

summaries. The discursive organization of a text implies the global structure of the text and it includes format of a document, threads of topics in the text and rhetorical structure of the text [6].

This approach asserts that texts are not just a linear sequence of clauses and sentences but they are a cluster of clauses and sentences, called discourse segments that are related pragmatically to form a hierarchical structure. Thus, in this approach, a text is first divided into discourse segments and based on these segments, the discourse structure of the text as intended by its author is re-constructed. This discourse structure or discursive representation of the text has been shown to be one way of determining the most important units of a text [20].

There have been various text summarization works that exploit the discursive representation of text. The most popular theory of text organization used for summarization has been the Rhetorical Structure Theory (RST) [9]. According to Mann and Thompson [21], one can associate a rhetorical structure tree to any text. RST is a binary tree representing rhetorical relations between text units. The relations tie together two non-overlapping pieces of text spans: the nucleus which is central to the writer‟s goal, and a satellite which is less central or a marginal part [7].

Rhetorical relations reflect semantic, intentional, and textual relations that hold between text spans. These text spans could be clauses or sentences extracted from the original text. Text spans could be related in RST in such a way that one text span may elaborate on another text span or one text span may provide background information for another text span [37]. In one application of discourse structure for text summarization importance score is associated to each clause in a text; the closer a clause is to the root of the tree, the higher is the score. Then clauses of highest score are extracted to produce a summary [20].

2.2 Text Representation Using Graphs

In this section, the concept of graph, representation of text using graphs and ranking algorithms that are used to identify important textual units in a graph are described.

Graphs

A graph is generally defined as G = (V, E) where V is the set of vertices or nodes and E is the set of edges connecting vertices in V. An edge in the graph is defined as e = (u,v) where u and v are

vertices in V. There are generally two types of graphs: directed and undirected. An undirected graph is one in which the pair of vertices in any edge in the graph is unordered, that is (v0, v1) = (v1, v0) and a directed graph is one in which each edge is a directed pair of vertices, that is (v0, v1) ≠ (v1, v0). The total number of edges incident to a given vertex defines the degree of the vertex and vertices in a directed graph can have two kinds of degree: in-degree and out-degree. The in-degree of a vertex v is the number of edges that have v as the head and the out-degree of a vertex v is the number of edges that have v as the tail. Furthermore, edges in a graph can have weights associated with them to give a weighted graph [44, 45].

Graphs are usually represented as an adjacency matrix which is an n by n matrix where n is the number of vertices in the graph. A particular cell in the matrix can have a value of 0 or 1 based on the existence of an edge between two vertices. The adjacency matrix for an undirected graph is symmetric whereas that of a directed graph need not be symmetric. Adjacency matrix representation of graphs makes the determination of links between vertices quite easy. For a directed graph, a row sum gives the out-degree of the vertex associated with that row while a column sum is the in-degree of the vertex. The degree of a vertex in an undirected graph is equal to either the row sum or column sum associated with the vertex [44, 45].

Graph-Based Ranking Algorithms

Graph-based ranking algorithms decide the importance of vertices within a graph by taking into account global information recursively computed from the entire graph rather than relying on only local vertex-specific information. The basic idea of ranking algorithms is based on the concept of prestige in social networks which has also been the source of many ideas in computer networks and information retrieval. A social network is a mapping of relationships between interacting entities such as people, organizations, computers, etc. Social networks are represented as graphs where the nodes represent the entities and the links represent the relations between the nodes. Prestige in social networks refers to the importance of nodes in the graph [47, 48].

Graph-based ranking models compute the prestige of a node within a graph based on the idea of "voting" or "recommendation". That is, when one vertex links to another one, it is basically casting a vote for that other vertex. The higher the number of votes that are cast for a vertex, the higher is the importance of the vertex. However, in many types of social networks, not all of the votes that

are cast for a vertex are considered equally important. The most notable ranking algorithm that is based on this fundamental idea is PageRank [47, 48].

PageRank is one of the most popular ranking algorithms originally designed as a method for Web link anlysis and still serves as the underlying mechanism behind the Google search engine. It is a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. Thus, PageRank can be thought of as modeling the behavior of a "random surfer" on the Web. The "random surfer" simply walks on the entire Web for an infinite number of steps. However, the surfer will occasionally get bored and instead of following a link pointing outward from the current webpage, he/she will jump to another random webpage. The probability of the surfer visiting a webpage is then proportional to its PageRank score [48, 49, 50].

Mathematically, the PageRank of a given page in the Web is calculated using the following formula [48]:

$$p(u) = \frac{d}{N} + (1-d)\sum_{v \in In(u)} \frac{p(v)}{Out(v)} \tag{2.3}$$

where, p(u) is the PageRank of a page u or node u,

N is the total number of nodes in the graph,

d is a parameter which is typically chosen in the interval [0.1, 0.2],

In(u) is the set of nodes that link to node u,

Out(v) is the out-degree of node v, and

p(v) is the PageRank of node v.

Another graph-based ranking algorithm is HITS (Hyperlink-Induced Topic Search) [51]. It is an iterative algorithm that is designed for ranking Web pages according to their degree of "authority". Unlike PageRank, it computes two sets of scores for each page: an "authority" score and a "hub" score. An "authority" score of a page estimates the value of the content of the page based on the number of incoming links to the page. A "hub" score of a page estimates the value of its links to other pages. More specifically, "authority" and "hub" scores are calculated as follows:

$$HITS_A(v_i) = \sum_{v_j \in In(v_i)} HITS_H(v_j) \quad (2.4)$$

$$HITS_H(v_i) = \sum_{v_j \in Out(v_i)} HITS_A(v_j) \quad (2.5)$$

where, $HITS_A(v_i)$ is the "authority" score of a node v_i,

$HITS_H(v_j)$ is the "hub" score of node v_j,

$In(v_i)$ is the set of nodes that link to node v_i, and

$Out(v_i)$ is the set of nodes that node v_i links to.

The algorithm computes "hub" and "authority" scores by first assigning arbitrary values to each node in the graph and iterates the computation until convergence (the difference between consecutive values) below a given threshold is achieved [51, 52].

Adaptations to PageRank and HITS

The original definitions of graph-based ranking algorithms assume unweighted graphs. This is based on the assumption that in the context of Web surfing or citation analysis, usually a vertex does not include multiple or partial links to another vertex. However, when graphs are built for natural language texts, there may be multiple or partial links between textual units. Hence, it is useful to integrate into the original PageRank and HITS model the "strength" of the connection between two vertices. That is both PageRank and HITS are redefined for a weighted graph. For instance, when the PageRank formula is modified to integrate edge weights into the graph, equation 2.3 becomes [46, 48]:

$$p(u) = \frac{d}{N} + (1-d) \sum_{v \in In(u)} \frac{w(v,u)}{\sum_{z \in Out(v)} w(z,v)} p(v) \quad (2.6)$$

Where, w(v, u) is the weight of the edge connecting node v and u which represents the strength of the connection between the nodes.

Similar adjustments can also be made to compute "hub" and "authority" scores in a weighted graph.

Furthermore, the PageRank algorithm has also been modified to be topic-sensitive. The basic model of PageRank assumes a random walker on the hyperlink graph jumps from the current node

to any node with fixed probability. However, it is possible to restrict the random walker to jump only to a random node which has non-zero similarity with the query which gives a topic-sensitive PageRank [57]. Topic-sensitive weighted PageRank is computed using the following formula:

$$p(u) = d\frac{sim(u,q)}{\sum_{y \in S} sim(y,q)} + (1-d)\sum_{v \in In(u)} \frac{w(v,u)}{\sum_{z \in Out(v)} w(z,v)}\, p(v) \quad (2.7)$$

where, S is the set of all nodes in the graph and

sim(u, q) is the similarity score between node u and the query q.

2.3 Information Retrieval Models

In this Section, the basic concepts of two information retrieval models: Vector Space Model (VSM) and Latent Semantic Analysis (LSA) are discussed. Considering its relevance to this thesis, LSA is described in detail in this Section.

2.3.1 The Vector Space Model

The Vector Space Model (VSM) is one of the oldest and most extensively studied models for text mining. In this model, a set of documents are conceptualized as a two-dimensional co-occurrence matrix, where the columns represent the documents and the rows represent the unique terms (usually words or short phrases) occurring in the documents. The value in each cell of the matrix reflects the importance of the term in representing the semantics of the document. Typically, the value in a particular cell is a function of the frequency with which the term occurs in the document. This value is usually adjusted with information retrieval weighting algorithms so that it reflects the importance of the term within the document more properly [2].

Similarity in the Vector Space Model is determined by using associative coefficients based on the inner product of pairs of documents where word overlap indicates similarity. Several mathematical measures of vector similarity have been proposed in the literature and of these, the most popular measure is the cosine similarity. For instance, when used in information retrieval task, cosine similarity measures the angle between the document vector and the query vector.

Hence, the documents returned as a response to a user query are those geometrically closest to the query according to the value obtained using the cosine similarity measure. More specifically, cosine similarity between two documents d_1 and d_2 is calculated as [2, 23]:

$$sim(d_1, d_2) = \frac{V(d_1).V(d_2)}{\|V(d_1)\| \, \|V(d_2)\|} \tag{2.8}$$

where $v(d_1)$ and $v(d_2)$ are the vector representations of d_1 and d_2 respectively, the numerator represents the dot product of the vectors while the denominator is the product of their Euclidean lengths. The effect of the denominator is to length-normalize the vectors.

Despite its success, the Vector Space Model suffers from the problem of literal term mismatch. That is, unrelated documents may be retrieved simply because terms occur accidentally in it, and on the other hand related documents may be missed because no term in the document occurs in the query [23]. This is usually referred in information retrieval as the problem of synonymy- the possibility of expressing a given concept in many ways and polysemy-the fact that most words have multiple meanings. Hence, this problem led to the development of new methods building on VSM and of these, the best known is Latent Semantic Analysis (LSA) or also known as Latent Semantic Indexing (LSI)[1] [2].

2.3.2 Latent Semantic Analysis

LSA, as defined in [24], is a fully automatic mathematical/statistical technique for extracting and inferring relations of expected contextual usage of words in passages of discourse. Its basic assumption is that the aggregate of all the word contexts in which a given word does and does not appear provides mutual constraints that determine the similarity of meanings of words and sets of words to each other. LSA tries to overcome the problem of literal term mismatch by allowing retrieval to be based on concepts rather than on terms. Unlike other natural language processing techniques, LSA uses no humanly constructed dictionaries, knowledge bases, semantic networks, grammars, syntactic parsers, morphologies, or the like. It is solely based on a collection of documents separated into words or meaningful terms.

[1] LSA and LSI are basically the same and are usually used interchangeably in the literature.

Another key assumption of LSA is that every document has an underlying semantic structure that can be captured and quantified in a matrix. To this end, LSA estimates the meaning of a word as a kind of the average of the meaning of all the sentences in which it appears, and the meaning of a sentence as a kind of the average of the meaning of all the words it contains [2, 24].

Though LSA is based on the Vector Space Model, it extends the model in a very important way. Specifically, it projects documents into a space with "latent" semantic dimensions. Moreover, the latent semantic space that the document is projected into has fewer dimensions than the original space. Latent semantic analysis is thus a method for dimensionality reduction. A dimensionality reduction technique takes a set of objects that exist in a high-dimensional space and represents them in a low dimensional space, often in a two-dimensional or three-dimensional space for the purpose of visualization [2, 24].

LSA exploits Singular Value Decomposition (SVD) to achieve dimensionality reduction. SVD is a very famous theorem in linear algebra which asserts that any real-valued rectangular matrix can be represented as the product of three smaller matrices of a particular form [63]. More specifically, the SVD of an m × n rectangular term-sentence co-occurrence matrix (A) is defined as:

$$A = U\Sigma V^T \tag{2.9}$$

where, $U = [u_{ij}]$ is an m×n column-orthonormal matrix whose columns are the eigenvectors of the matrix AA^T and they are called left singular vectors.

Σ is an n×n diagonal matrix, whose diagonal elements are non-negative singular values[2] sorted in descending order.

$V = [v_{ij}]$ is an n×n orthonormal matrix, whose columns are the eigenvectors of the matrix A^TA and they are called right singular vectors.

Thus, SVD maps each column vector of A, which represents the weighted term-frequency vector of a sentence, to the columns of vector V^T, and maps each row vector of matrix A, which represents the number of times a term occurs in each of the sentences in a document, to row vector of matrix U [25].

[2] The singular values are the (positive) square roots of the eigenvalues of AA^T or A^TA. These eigenvalues are positive real numbers, because AA^T is symmetric and positive definite.

Once the SVD of a term by sentence matrix is computed, the next task is to find a low-rank approximation to the matrix. This can be done by keeping only the "k" largest singular values of Σ along with the corresponding columns in the matrices U and V^T resulting in matrices U_k, Σ_k, Vk^T, and A_k . The new matrix A_k has the same number of rows and columns as that of the original matrix but it is only approximately equal or a least square best fit to the original matrix A. The columns of U_k define the topics of a document, with the rows representing terms of the document. Similarly, the rows of V_k^T define the topics of a document, with the columns representing sentences of the document [7, 13, 54]. Figure 2.1 illustrates the Singular Value Decomposition of a term by sentence matrix [13].

The main reason of approximating the original matrix is to reduce the noise in the original term by sentence matrix which is caused by polysemy and synonymy. Besides, the reduced matrix is capable of capturing the hidden semantic structure in the document represented by matrix A. That is, interrelationships among terms can be captured which allows for identification of semantically similar documents that share few or no common terms. For instance, synonym words such as doctor and physician may not co-occur in a given document but they generally appear with closely related concepts such as hospital, medicine, nurse, etc. Because of the conceptual similarity between the words doctor and physician, the words will be mapped near to each other in the reduced matrix [25].

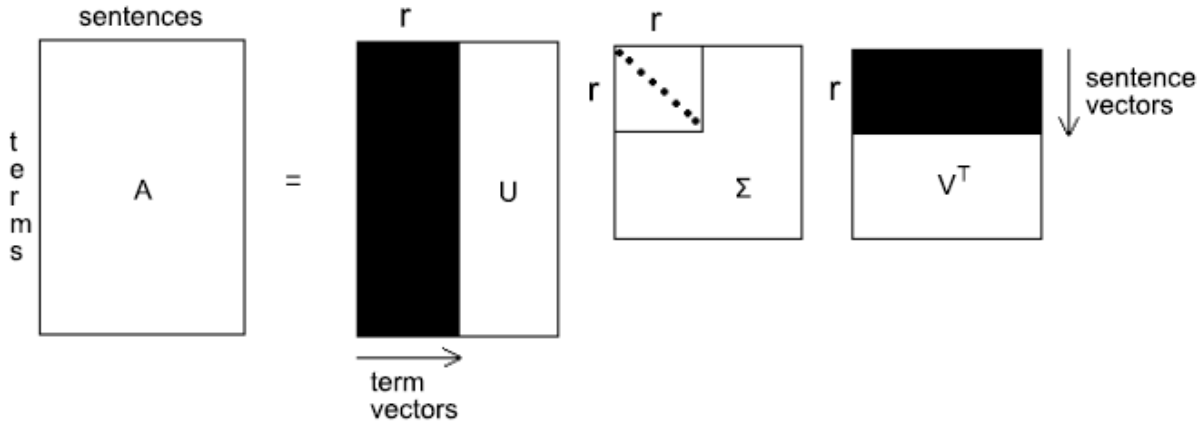


Figure 2.1: Approximate reconstruction of a matrix

However, there is one big problem in using SVD for dimensionality reduction which is the selection of the right dimensionality or the value of „k". If the value of "k" is too large, it

renders the matrix too noisy to be useful whereas if the value of “k” is too small, the reduced matrix will fail to capture the real semantic structure in the documents. Therefore, the optimal dimensionality must be determined empirically [2].

Query Representation

When LSA is applied in information retrieval, a user‟s query is often considered as a pseudo-document and is represented as a vector in the reduced term-by-document space. First, the terms in the query are represented by an (m X 1) vector „q‟ whose elements are either zero or the frequency of the terms that exist in the database of the reduced vector space. The local and global term weights used for the document collection are also applied to each non-zero element of the query vector q. Then the query vector is represented in the reduced LSA space by the vector q' which is calculated as follows [54]:

$$q' = q^T.U_k .\Sigma_K^{-1} \tag{2.10}$$

where $q^T.U_k$ is the sum of term vectors specified by vector q scaled by Σ_K^{-1}.

Thus, the query vector is represented by the weighted sum of its constituent term vectors. The query vector can then be compared to all existing document vectors and the documents are ranked according to their similarity (nearness) to the query. The most commonly used similarity measure in LSA is the cosine between the query vector and the document vector [54].

Sentence Similarity Matrix

As described above, LSA assigns terms to topics and topics to sentences. This assignment of topics to sentences depends on the similarity between sentences. Similarities between two sentences can be computed using the columns of the term by sentence matrix since each column of the term by sentence matrix represents a sentence. Thus, if T is a square matrix containing all the similarities between sentences in a given document and matrix A is the SVD of a term by sentence matrix, then T can be calculated as follows [58]:

$$T = A^T A$$

$$= (U\Sigma V^T)^T U\Sigma V^T$$

$$\approx (U_k \Sigma_k V_k^T)^T U_k \Sigma_k V_k^T$$

$$= V_k \Sigma_k U_k^T U_k \Sigma_k V_k^T \quad = \quad (V_k \Sigma_k)(V_k \Sigma_k)^T \qquad (2.11)$$

Thus the rows of the matrix $V_k\Sigma_k$ can be taken as vectors representing the sentences and the dot product between the rows of the $V_k\Sigma_k$ gives the similarity between sentences. This gives the advantage of computing sentence similarity based on their semantics rather than computing similarity based on mere common word occurrences. Furthermore, it is to be noted that the rows of the matrix V_k^T are the eigen vectors of the similarity matrix T [58].

2.4 Evaluation of Automatic Text Summarization

In any research area where there are different approaches, evaluation is crucial to assess the achievements of these approaches and to identify the ones that are worth pursuing. In text summarization, there is no standard method for summary evaluation and research papers often use their own methods. However, we can generally distinguish between two types of evaluation methods: intrinsic and extrinsic. An intrinsic or normative evaluation is an assessment of the quality of the summary. It usually involves human judges who determine the quality of the summary by directly analyzing it. Text quality refers to some aspects of the text such as grammaticality, non-redundancy, reference clarity and coherence. An intrinsic evaluation could also be a measure of how well the summary compares with an ideal summary written by the author of the source text or a human abstractor [13, 30].

An extrinsic or task-based evaluation aims at measuring the performance of using the summaries for a certain task. That is, the quality of a summary is judged by users according to how it influences the achievement of some other task, such as how well it helps them determine the source‟s relevance to topics of interest or how well they can answer certain questions relative to the full source text [30].

However, both intrinsic and extrinsic evaluation methods are not entirely satisfactory. The main problem with intrinsic approach is the difficulty of constructing a unique ideal summary for a given document or a set of documents. As there are many ways to describe an event or a scene, it is possible that users can produce more than one summary to a particular document. Hence,

agreement between human judges becomes an issue. Besides this, manual evaluation is too expensive. Extrinsic evaluation, on the other hand, is time-consuming, expensive and requires a considerable amount of careful planning [31, 32]. Various approaches to intrinsic and extrinsic evaluation methods can be found in the literature and here, we mention two intrinsic methods: co-selection and content based measures.

2.4.1 Co-Selection Measures

Most summarization systems select the most representative sentences in the input to form an extractive summary. In such settings, the quality of the summary is usually determined using co-selection measures which find out how many of the sentences in the automatic summary are contained in the ideal or manual summary. Precision, recall, and F-Score, which are the commonly used information retrieval metrics, are the main evaluation metrics of co-selection. Precision (P) is the ratio of the number of sentences occurring in both system and ideal summaries to the number of sentences in the system summary. Recall (R) is the ratio of the number of sentences occurring in both system and ideal summaries to the number of sentences in the ideal summary. F-Score (F) is a composite measure that combines precision and recall. Basically, F-Score is calculated as follows [13, 31]:

$$F = \frac{2*P*R}{P+R} \tag{2.12}$$

The advantage of using co-selection measures is that once human judges define the gold-standard summary, it can be repeatedly used to evaluate automatic summaries by a simple comparison. Unfortunately, there are also several disadvantages. The main problem, as discussed in the above section, is to define a gold-standard summary. It has been shown that the difference in recall measure of a summary may range from 25% up to 50% depending on which of two available human extracts are used for evaluation. Thus, using co-selection measures creates the possibility that two equally good extracts are judged very differently.

Many of the subsequently developed evaluation measures were designed to address the problems with precision and recall. For instance, it has been suggested that more emphasis be given to recall than precision. This is because precision might be too strict in that some of the sentences chosen by the system might be good though they have not been chosen by the gold-standard. However, recall

measures the overlap with already observed sentence choices. It has also been suggested to use multiple human judges rather than a single person‟s judgment [32].

The other evaluation metric that can be used in co-selection measures is relative utility and it was introduced as an improvement to precision and recall. The method involves multiple judges who score each sentence in the input text with confidence values for their inclusion in the summary. The principle is, thus, highly ranked sentences should have a high probability of being included in a summary and low ranked sentences should have very low or no probability of inclusion. Hence, each possible selection of sentences by a system can be assigned a score showing how good a choice of sentences it represents. Other than requiring a good deal of manual effort in sentence tagging, this approach offers a simple and easy way of evaluating summaries [13, 32].

2.4.2 Content-Based Measures

Co-selection measures can only determine the quality of a summary based on the number of sentences that are common to ideal and automatic summaries. However, two sentences can contain the same information even if they are written differently. This weakness of co-selection measures is addressed in content-based similarity measures. The advantage of using content-based similarity measures is that two summaries can be compared at a more fine grained level than just sentences. There are several content-based similarity measures that take into account different properties of the text such as cosine similarity, word overlap, longest common subsequence, ROUGE (Recall-Oriented Understudy for Gisting Evaluation) and LSA-based measures [13, 31]. Among these, cosine similarity and ROUGE are the most common ones in the literature. Here, we will briefly discuss ROUGE as we have already discussed cosine similarity in Section 2.3.

ROUGE is based on the computation of n-gram overlap between a summary and a set of ideal summaries. It takes as input pairs of auto–generated summaries and their corresponding ideal summaries, and determines their similarities based on different features [13, 32]. The use of a generally agreed on and automatic metric such as ROUGE allows cheap evaluation and ease in comparing results from different research efforts. Thus, ROUGE has become a de-facto standard evaluation method in the field of automatic summarization. For instance, Document Understanding Conference (DUC), which is a series of evaluation workshops held each year to evaluate summarization systems, has been using ROUGE since 2003 [13]. In this thesis, we use F-Score to

evaluate the performance of our summarization system. As can be seen in equation 2.12, F-Score gives the same importance to precision and recall, thus, using it as an evaluation measure is a good trade-off between precision and recall.

2.5 Summary

Automatic text summarization is a process by which the most important concepts in a document are identified and then presented in a condensed and human-readable form. In order to achieve this various researchers have proposed different summarization approaches. In this Chapter, we discussed summarization approaches by classifying them into three categories: surface level, entity level and discourse level. Almost all summarization algorithms begin with document preprocessing which includes activities such as tokenization, normalization, stop-word removal and stemming. Each of these activities was discussed in detail in this Chapter.

The most challenging task in text summarization is evaluation and as a result various researchers have proposed different evaluation techniques. Among these evaluation methods co-selection and content-based measures were reviewed as they are the most widely used in the literature. Theoretical backgrounds of LSA and graph-based ranking algorithms were also presented as the proposed summarization system in this thesis is based on them.

CHAPTER THREE: RELATED WORK

There has been a great amount of research in the field of automatic summarization in recent years and of these, the most notable are those based on algebraic reduction methods. Algebraic reduction methods are purely lexical approaches that work only with the context of terms and thus they do not rely on a particular language. There are several algebraic reduction methods and some of them are Latent Semantic Analysis (LSA), Probabilistic Latent Semantic Indexing (PLSI), Non-negative Matrix Factorization (NMF) and Semi-Discrete matrix Decomposition (SDD). However, the most widely used is LSA [7].

In this Chapter, we will critically review previous summarization works based on latent semantic analysis and graph-based ranking algorithms. We will also review and identify gaps that exist in previous summarization works for Amharic language. Finally, we will identify the gaps to be bridged by our work.

3.1 LSA-Based Automatic Text Summarization

The motivation to apply LSA for automatic summarization stems from the assumption that each of the resulting right singular vectors in SVD decomposition of a term by document matrix represents a salient topic or concept of the document. The importance of a salient topic or concept is indicated by the magnitude of the corresponding singular value and the sentence that best represents this topic or concept will have the largest value with this vector [25].

Gong and Liu [25] are the first to propose a scheme for automatic text summarization using LSA. Their LSA based approach classifies the document into different topics and picks the dominant sentence for each topic until the desired summary length is reached. Their algorithm can be stated as follows:

1. Decompose the document D into individual sentences, and use these sentences to form the candidate sentence set S, and set k = 1.
2. Construct term by sentence matrix A for the document D.

3. Perform SVD on A to obtain the singular value matrix Σ, and the right singular vector matrix V^T. In the singular vector matrix, each sentence i is represented by the column vector i = [vi1 vi2 …vir] ^{T} of V^T.
4. Select the k^{th} right singular vector from matrix V^T.
5. Select the sentence which has the largest index value with the k^{th} right singular vector, and include it in the summary.
6. If k reaches the predefined number, terminate the operation; otherwise, increment k by one, and go to Step 4.

The main assumption in Gong and Liu‟s approach is that the rows of V^T are regarded as defining topics with the columns representing sentences from the document. Hence, summarization is equivalent to finding the sentence with the highest value for each row in V^T. The summary produced using this approach is generic since it selects sentences describing different topics. Furthermore, the summary produced by this method contains the minimum redundancy as there is no correlation between any two singular vectors. In mathematical terms, such vectors are said to be linearly independent [2].

Steinberger [13, 62] has also proposed a text summarization approach which is based on LSA. The author‟s work is in an attempt to solve the main drawback of Gong and Liu‟s approach. Gong and Liu‟s approach selects one sentence for each topic and due to this, important sentences may not be included in the summary. That is, when k sentences are extracted, the top k topics are treated as equally important. Thus, the summary may include sentences about less important topics. To address this problem, the author has proposed a new criterion of selecting sentences that should be included in the summary. A summary now contains sentences whose vectorial representation in the matrix $\Sigma \cdot V^T$ has the greatest „length‟. More specifically, each sentence is assigned an SVD-based score using the following formula:

$$Sc_k^{SVD} = \sqrt{\sum_{i=1}^{n} v(i,k)^2 * \sigma(i)^2} \tag{3.1}$$

where, v(i, k) is the i^{th} element of the k^{th} sentence vector,

σ(i) is the corresponding singular value, and

n is the number of dimensions of the reduced space.

Thus, those sentences with the highest Sc_k^{SVD} are selected to be included in the summary. This method makes it possible to choose sentences with greatest combined weight across all topics rather than always choosing one sentence for each topic as done in [25].

Murray *et al*. in [26] also employ LSA for extractive summarization of meeting records based on the framework presented in [25]. They address the same problem identified in [13] and another problem of Gong and Liu‟s approach. As described by the authors, Gong and Liu‟s approach ties dimensionality reduction to summary length. In order to solve these problems, the authors proposed a method which chooses the n best sentences for each topic with n determined by the corresponding singular value from matrix Σ. For instance, the number of sentences selected from the first topic is determined by the percentage that the largest singular value represents out of the sum of all singular values, and so on for each topic. Hence, dimensionality reduction is no longer tied to summary length and more than one sentence per topic can be chosen.

Another summarization method which employs LSA is that of Jagarlamudi *et al*. [35]. The authors use the rows of $V_k.\Sigma$ to represent the sentences (S_i) in lower dimension space. Since their system is query-oriented, user‟s information need or query is also projected into latent dimensions using $Q = Q^T.U_k.\Sigma_k^T$. The relevance score of the reduced sentences (S_i) are then computed by their cosine similarity measure with the new query vector and the top ranking sentences are then concatenated to generate a summary.

Yeh *et al*. in [28] have also proposed a summarization method that uses LSA. The method proposed makes use of LSA to find the semantic representation of sentences which is used as an input for creating a text relationship map. In order to find the semantic representation of sentences, they performed SVD on a word by sentence matrix, reduced the dimensionality of the latent space and reconstructed the corresponding matrix $A= U_k\Sigma_k (V_k)^T$ where the columns of A are semantic sentence representations. Text relationship map is then created where two sentences are connected in the map if the cosine similarity of their semantic representation is above a pre-defined threshold. The significance of a sentence is then measured by counting the number of links it has in the text relationship map. Finally, the top scoring sentences are selected from the map to generate a summary.

3.2 Graph-based Ranking Algorithms for Extractive Summarization

Ranking algorithms such as HITS and PageRank have successfully been used in Web-link analysis and social networks. More recently, graph-based ranking algorithms have also been used in text processing applications, particularly for text summarization. Graph-based ranking algorithms are used in text summarization based on the assumption that sentences that have more relationships with other sentences in a given text are more important because they can directly relate to more other sentences. In other words, the more relationships the sentences in the graph have, the more important they are.

Erkan and Radev in [48] proposed a summarization system based on graph-based centrality scoring of sentences. The proposed system makes use of a similarity graph where the cosine similarity of each pair of sentences is computed. Sentences are then ranked based on their centrality score in the graph. The authors have introduced three different methods for computing centrality score in the similarity graph. The first and the simplest approach is degree centrality which is defined as the in-degree of vertices after removing edges which have cosine similarity below a pre-defined threshold.

The second approach for computing centrality score is based on a modified PageRank model called LexRank. LexRank, unlike the original PageRank method, applies random walk on undirected similarity graph after removing edges below a pre-defined threshold. The third approach is called continuous LexRank and is similar to the second approach except that it applies random walk on a weighted and fully connected similarity graph. This is in an effort to solve the problem of information loss incurred in the first and second approaches due to the removal of some edges in the graph. The results of applying these methods on extractive multi-document summarization have been shown to be quite promising.

Mihalcea and Tarau in [46, 47] have also proposed another similar graph-based random walk model called TextRank for extractive summarization. TextRank, unlike LexRank, considers three different ways of representing a sentence similarity graph. That is, similarity graphs can be represented as: simple undirected graph, directed weighted graph with the orientation of edges set from a sentence to sentences that follow in the text, or directed weighted graph with the orientation of edges set from a sentence to previous sentences in the text. Similarity between two sentences in

the graph is measured by the number of common tokens between the lexical representations of the two sentences. Furthermore, the study in [46] uses both weighted PageRank and weighted HITS to determine the importance of a sentence.

TextRank has been applied for both single document and multi-document summarization for Portugese and English languages and the results obtained are comparable with those of state-of-the-art summarization systems. In conclusion, the study in [46] has shown that both weighted ranking algorithms give better results when applied on graphs containing only backward edges.

3.3 Automatic Text Summarization Systems for Amharic

Though the field of automatic text summarization has enjoyed a lot of research for many languages, Amharic language and in general local languages are under represented in the area. However, few researchers have attempted to develop automatic summarization system for Amharic. Such efforts are reviewed below giving due attention to the techniques employed.

The first automatic text summarization work for Amharic is that of Kemal [36]. Kemal proposed a summarization system for Amharic news items based on the extraction approach. The proposed system has two basic phases: extraction and learning. In the extraction phase, the system applies sentence weighting formula to assign weight to each sentence in the text. Different statistical features including presence of title words, cue phrases and keywords are used to compute the weight of a sentence. The system then selects the top scoring sentences to form a summary.

In the learning phase, the system tries to update some of the features it uses to weight sentences. This is an attempt to make the system dynamic as the author believes that using specific number of features makes the system static and limits its performance to a fixed level. Basically what this phase does is that it accepts a list of stop words and cue phrases from the user, appends this to its database of stop words and cue phrases, and makes use of this for an improved sentence weight calculation.

The system is also trained with four news articles which have a corresponding manual summary. This is to find the appropriate contribution of each statistical feature towards the importance of a sentence and to adjust the weight of each feature accordingly. Evaluation results have revealed that

the use of title words and more number of keywords are important to create a summary containing the core points of the news text.

Teferi [33] tried to apply a machine learning technique called Naïve Bayes to automatic summarization of Amharic news texts. The system developed by Teferi has two phases: training and testing. To train the system, 480 Amharic news articles with their corresponding manual summaries were prepared. In the training phase, the Naïve Bayes classifier is trained to identify summary-sentences from non summary-sentences based on four feature values. Feature values used were the presence of title words, location of sentences in the document, presence of cue words and presence of thematic or highly frequent words. Thus, the sole aim of the training phase is to discover a classification function that accepts a sentence as an input and outputs the probability of the sentence being included in a summary based on the four feature values.

After the training phase, the performance of the classifier in determining a sentence as a summary-sentence was measured based on classification success rate, precision and recall. For this purpose, 20 documents not in the training dataset along with their manual summaries were prepared. Furthermore, evaluation results have shown that the use of multiple features as a combination yields better results in predicting summary sentences.

Another research work on summarization for Amharic document is that of Helen [34]. Different from the previous two works, Helen‟s work is on the domain of Amharic legal judgment documents. However, the techniques used are similar to those used in Kemal‟s work. More specifically, the summary generation begins with a manual segmentation of the given document to detect its themes. Five legal themes were identified and these were introduction, reason, fact, judicial analysis, and decision. Then, each sentence in a given document is weighted based on two features: cue phrases and sentence location. Finally, sentences with the highest weight are selected at 20% compression rate from each segment and are merged together to serve as a single summary for the document.

For the purpose of evaluating the system, the author has prepared two summaries for each test document: ideal or manual summaries and random summaries. Random summaries are extracts that are automatically generated simply by taking n number of sentences from the original document. Evaluation is then performed by calculating the precision and recall of both the system summary

and the random summary against the manual summary. In conclusion, evaluation results have shown that system summaries are much closer to manual summaries than are random summaries.

Daniel [55] has applied traditional statistical methods to summarize single document Amharic news texts. After pre-processing each news article, the system proposed computes the significance of a sentence based on four statistical features. These are position of a sentence, frequency of numeric terms in a sentence, summation of term frequency of every word in a sentence and the number of title words in a sentence. Furthermore, sentences are normalized to reduce the probability of favoring long sentences in the summary. The top ranking sentences are then extracted to make a summary. To evaluate the system proposed, the author has prepared 30 pairs of manual and system summaries for Amharic news articles. System summaries were then evaluated objectively and subjectively. In the objective evaluation of the system, the precision and recall of the system summaries are computed against their corresponding manual summaries. In the subjective evaluation, on the other hand, the linguistic qualities of the system summaries are assessed by the same individuals who prepared the manual summaries. Finally, evaluation results have shown that the system performs better when position feature alone is used which is in contrary to the results obtained in [33].

Multi-document summarization for Amharic news texts has also been developed by Abraham [56]. Abraham, like previous works in Amharic news summarization, used pure statistical approaches to extract sentences from multiple documents. After pre-processing each news article, the system proposed computes the significance of a sentence based on four statistical features: Context Sensitive Frequency Based, number of title words in a sentence, centroid score, and position of a sentence in a text. Context Sensitive Frequency Based feature scores sentences based on the average of the summation of the probability distribution of terms in the sentence. Probability distribution of a term is the quotient of the term"s frequency in the event set and the total number of terms in the event set.

As described in the paper, the centroid score of a sentence in the document cluster or eventset is computed based on the cosine similarity value between the sentence and the cluster centroid. The centroid of a cluster is represented by a vector of terms associated with their corresponding average TFIDF value. Once the sentences are scored based on the four features and the combination of all

these features, the top ranking sentences are taken as a summary at a 20, 30 and 40% compression rates. For the purpose of evaluation, Abraham prepared 60 news items collected from three Amharic news providers. These news items are grouped into 20 event sets or clusters in which each event set consists of three news texts about one common topic.

Manual and system summaries were then produced for each event set and the system summaries were evaluated objectively and subjectively. In the objective evaluation, precision and recall were used and the result indicated that summarization based on Context Sensitive Frequency Based feature alone and summarization based on occurrence of title words feature alone performed well. In the subjective evaluation, the redundancy and the linguistic quality of the system summary were assessed and the results were shown to be promising. Similarly, Winta [59] has applied a machine learning tool called WEKA for Amharic multi-document summarization. In addition to the feature values used by Teferi, the author has used centroid score, sentence length cut-off and TFIDF to train the machine learning tool.

3.4 Summary

This chapter reviewed different text summarization systems that are related to our study either in summarization approach or language behavior. The review has shown that LSA and graph-based ranking algorithms can successfully be used for extractive summarization. Summarization using LSA is based on the assumption that the right singular matrix obtained after the SVD decomposition of a term by sentence matrix captures the salient topics of a document. Summarization is then accomplished by selecting sentences that reflect these topics. However, the number of topics that will be included in the summary is a very important decision. The summary should include sentences that are about the important topics but it should omit those that are about unimportant topics.

In order to create a good generic summary, all the major topics present in the document should be considered. To accomplish this, different approaches have been proposed in previous LSA-based summarization works. We also use LSA in this thesis to generate a generic summary that contains sentences from all the major topics present in the document. However, unlike previous LSA-based summarization works which use the right singular vectors to identify semantically important sentences, we use the left singular vectors to construct a topic vector and extract sentences that are

relevant to this topic vector. The topics or concepts identified by the left singular vectors can be represented by a set of terms that have high index value in each left singular vector. These sets of terms are used in this thesis to construct a topic vector.

Thus, it is our assumption that by using terms to represent the topics of a document, a wide range of topics in a document can be covered. Hence, a summary generated by extracting sentences that are relevant to the identified topic will have a wide coverage of the document"s main content. This also provides the possibility to add more than one sentence about an important topic rather than choosing always one sentence for each topic as done in [25]. Furthermore, document genre information such as sentence resemblance to title and sentence position in a document is also considered in generating a summary. This is in an attempt to further improve the quality of the summary.

Graph-based ranking algorithms, on the other hand, represent a document using a graph in which a vertex is added for each sentence in the document and edges between the vertices are established using sentence similarities. Sentences are then ranked based on their importance in the graph and the top ranking sentences are extracted to generate a summary. The importance of a sentence is computed based on its similarity to other sentences in the document and hence, the similarity measure used has a profound effect on the summarization result obtained.

So far, summarization works that employ graph-based ranking algorithms use only the lexical representation of the sentences to compute their similarity. However, as described in Chapter 2, similarity measure using lexical representation is prone to synonymy and polysemy problem. In this thesis we attempt to solve the above problem by constructing sentence similarity graph using LSA which is discussed in detail in Chapter 2. Thus, we assume that applying ranking algorithms on a graph that is constructed based on the semantic similarities of sentences offers better result in text summarization. Furthermore, construction of graphs will be made to employ document title as a source of topic. This helps to penalize sentences that do not belong to the main topic of the document.

Finally, this Chapter also reviewed previous text summarization works for Amharic. So far all summarization works for Amharic texts are based on pure statistical methods and machine learning algorithms. Pure statistical methods fail to capture the main topics of a document as they are

ignorant of the semantics of the words in the document. Machine learning algorithms, on the other hand, require a great deal of training corpora and hence, they are very costly. Moreover, summarization systems based on machine learning algorithms are not easily adaptable to other languages or domains.

Thus, in this thesis we attempt to consider the contextual meaning of words to capture the main topics of a document and hence, summaries of high quality can be produced. In order to achieve this, we propose two different approaches. The first approach is based on LSA and document genre information, and the second approach is based on graph-based ranking algorithms. Furthermore, these approaches do not need any training corpora which make them easily adaptable to different languages and domains.

CHAPTER FOUR: DESIGN AND IMPLEMENTATION OF AUTOMATIC AMHARIC TEXT SUMMARIZER USING LATENT SEMANTIC ANALYSIS

This Chapter describes the design and implementation of automatic Amharic news text summarizer using Latent Semantic Analysis. The design and implementation process of the proposed summarizer consists of four major phases which are document preprocessing, semantic model analysis, sentence ranking, and sentence extraction. Pre-processing includes lexical analysis, normalization, stop-word removal, stemming, and term extraction. Semantic model analysis includes construction of term by sentence matrix, term weighting, Singular Value Decomposition and dimensionality reduction.

The third phase in the design and implementation process is sentence ranking. In this thesis we propose two sentence ranking approaches for summary generation. The first approach is based on topic identification using the left singular vectors in the Singular Value Decomposition of a term by sentence matrix. The second approach applies graph-based ranking algorithms on semantic sentence similarity matrix to determine the significance score of sentences. Finally, the sentence extraction module is responsible for extracting top ranking sentences to generate a summary. Detailed design and implementation of these modules is explained in this Chapter.

4.1 System Architecture

As described in the beginning of this Chapter, the automatic summarization system developed in this thesis has four different modules: preprocessing, semantic model analysis, sentence ranking, and sentence extraction. The general architecture of the system is shown in figure 4.1 and each module of the summarization system is explained in detail in the succeeding sections.

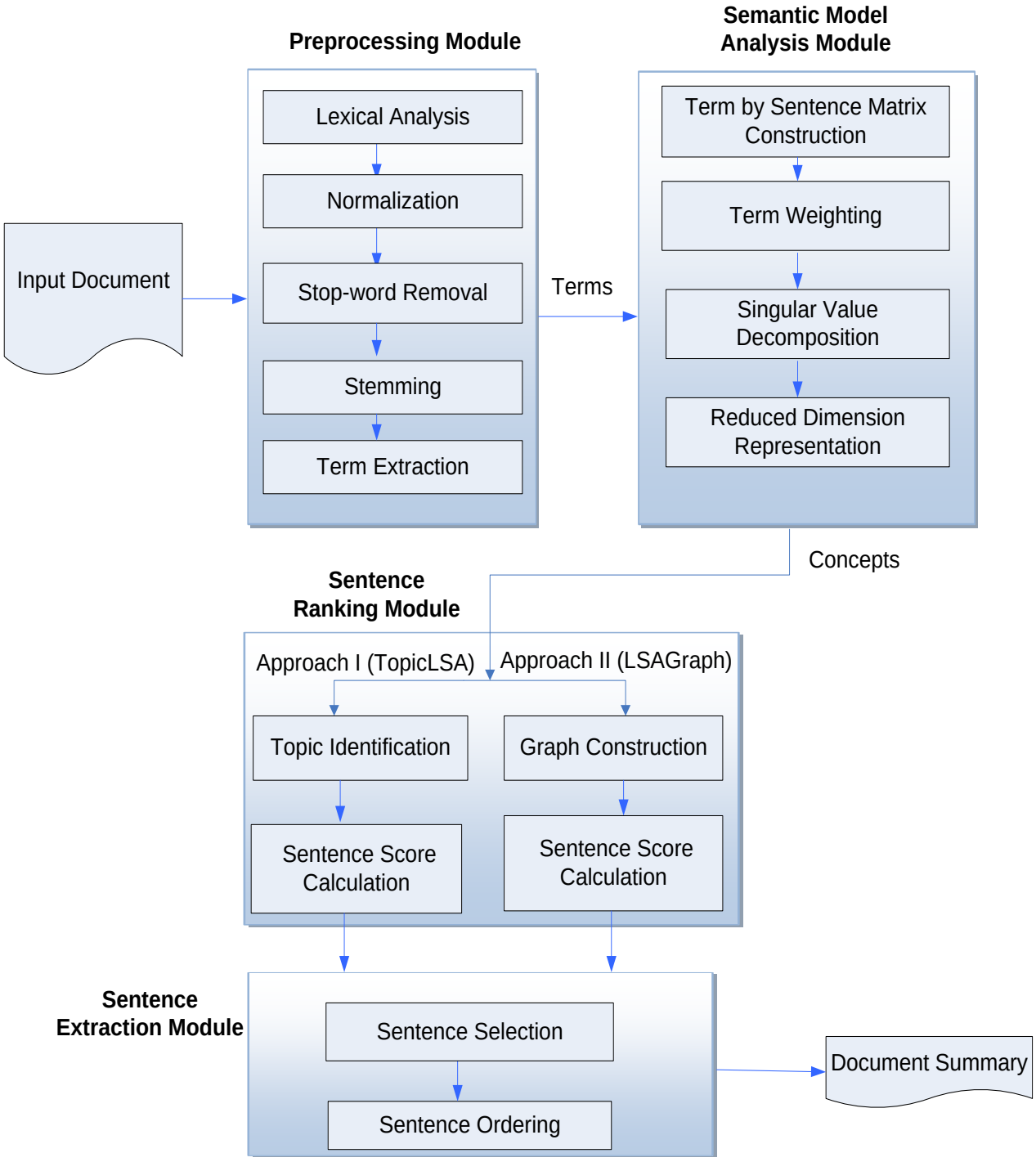


Figure 4.1: The general architecture of automatic Amharic text summarizer using LSA

4.1.1 Preprocessing Module

The first module of the summarizer is the preprocessing module. This module is responsible for producing a set of index terms that represent the input document. In order to achieve this, the module carries out lexical analysis, normalization, stop-word removal, stemming, and term extraction.

In this thesis, we have adopted the preprocessing module from the work of Tessema [41]. The preprocessing module of Tessema employs Lucene[3] which is a mature, free, open source, high performance, and scalable information retrieval library. It provides a simple Java API that allows different applications to integrate indexing and searching capabilities.

In the following sub-sections, detailed explanation of the adopted components is presented.

Lexical Analysis

The first step in the preprocessing of the input document is lexical analysis which is also known as tokenization. The generation of a summary depends on the computed score of each sentence, and these scores depend on the individual words that constitute the sentences. Hence, lexical analysis in text summarization involves text splitting into words and sentences. In this thesis, sentence extraction out of the text is based on the sentence delimiter, **አራት ነጥብ (።)**. Following the sentence splitting, the individual words are extracted from the sentences by scanning each sentence for predefined word delimiters such as new line, space, Amharic punctuations, etc.

Normalization

As explained in Chapter 2, there are several Amharic characters that have the same pronunciation and use, but different symbols. Such characters are automatically replaced by a common character in this component. For example, the different forms of the word „Hailu", which are **ሀይሉ**, **ሃይሉ**, **ሐይሉ**, and **ኃይሉ** are all converted to the common form **ሀይሉ** by changing the first character of the three words. Table 4.1 shows sample character replacements used by Tessema.

[3] http://lucene.apache.org

Table 4.1: Sample of Normalized Characters

Characters to be replaced	Replaced character
ሐ፣ኀ፣ኃ፣ሃ፣ሓ	ሀ
ዐ፣ዓ	አ
ሠ ሡ ⋮ ሦ	ሰ ሱ ⋮ ሶ
ኸ	ከ
ጎ	ገ

Furthermore, the normalization component includes the expansion of words that are written in a short form using "/" or "." into one word or two words. For example, **ጠ/ሚኒስትር** is expanded to **ጠቅላይ ሚኒስትር** and **ዶ/ር** is expanded to **ዶክተር**. To achieve this, each word obtained after tokenization is checked for its presence in a list of common short words and if a word is found to be in the list, it is expanded into its corresponding form. Around 40 short forms of single and compound words are considered in [41]. See Annex D for the list of short words and their expanded forms.

Stop-word Removal

Some words, referred to as "stop-words" are either words that do not contribute significantly to the overall topic of the document such as conjunctions, articles, pronouns or are words that appear in many sentences, and thus do not serve to topically distinguish one sentence from another. Such words can be identified and removed by using a predefined list of stop-words [39].

In addition to the common stop-words such as **ነው**, **ናቸው**, **ነበረ**, etc., there are also news specific stop-words such as **አስታውቀዋል**, **አመልክተዋል**, **ገልጸዋል**, etc**.** In this thesis, we have used around 150 common and news specific stop-words which are taken from previous studies [41, 59]. See Annex E for the list of stop-words.

Stemming

Once the process of removing stop-words is completed, the next task in the preprocessing of the input document is stemming. As described in Chapter 2, in this thesis, we employ the stemming algorithm developed in [41]. The stemming algorithm developed in [41] removes those affixes that are usually used for changing the tense, number, gender and case of a word. Furthermore, in the

case of removing suffixes with vowels, the last character of the word after the removal of the suffix is changed to *sades* (the six order of a character). A total of 33 suffixes and 17 prefixes are used in [41]. See Annex C for the list of suffixes and prefixes.

Term Extraction

As a result of completing lexical analysis, normalization, stop-word removal, and stemming, we now have an index of terms that are capable of representing the content of the document. These terms can further be refined to retain only those terms that occur above a certain threshold in the document. However, since Latent Semantic Analysis works on the idea of word co-occurrences or on the co-relation of words in the document, it is necessary to retain as many of the words in the document as possible. Hence, in this thesis, we have decided to use all terms in the document following the removal of stop words.

4.1.2 Semantic Model Analysis Module

This module begins with constructing a term by sentence matrix which is to be followed by term weighting, Singular Value Decomposition, and Dimensionality reduction. Each of them is discussed below.

Term by Sentence Matrix Construction

Latent Semantic Analysis, being a variant of Vector Space Model, requires both the documents and the queries (document title in our case) to be represented mathematically as vectors in some vector space [23]. To achieve this, a Java code is written that first identifies unique terms from the output of the first module and computes the frequency of each unique term in the sentences they are contained in. Then a term by sentence matrix is constructed which has as many rows as the number of unique terms in the document and as many columns as the number of sentences in the document. Thus, the entry at row m and column n of the matrix is filled with the frequency of the term m in sentence n. Document titles are also represented by one column in the term by sentence matrix using the database of terms identified in the document. We have used JAMA[4], a Java library

[4] http://math.nist.gov/javanumerics/jama

package, for constructing matrices and for all computations associated with matrix. JAMA is chosen because it is a free Java library package that provides SVD computation of a matrix.

Term weighting

The term by sentence matrix, which we constructed above, uses only local information of each term. That is, the weight of term i in sentence j is defined as a local weight (L_{ij}) where $L_{ij} = tf_{ij}$ and tf_{ij} is term frequency or the number of times term i occurs in sentence j. However, this way of representing terms has some serious drawbacks. One limitation is that long sentences are favored for inclusion in a summary simply because they tend to have more words, not because they are relevant. Furthermore, the relative global frequency of terms across the entire document is ignored and thus, it is not possible to determine the importance of each term in describing the topics of the document.

Thus, it is necessary to represent terms according to their distribution in the entire document. There are several weighting functions in the literature and of these the most common local and global weighting functions were discussed in Chapter 2. In this thesis, we have experimented with 16 combinations of local and global weighting functions to find the best weighting function for the proposed summarization approaches. The effect of these weighting functions on the performance of the proposed summarization approaches is described in Chapter 5.

Singular Value Decomposition

The next step in the semantic model analysis module is to compute the singular value decomposition of the weighted term by sentence matrix. The SVD of the weighted matrix is performed using a built in function of JAMA. The function accepts the weighted matrix as an input and returns three matrices: U, a term by dimension matrix, $\sum$, a singular value matrix, and V, a sentence by dimension matrix. As described in Chapter 2, for a term by sentence matrix A, the columns of the matrix U, which are also called left singular vectors, are the eigenvectors of the term similarity matrix AA^T and the columns of the matrix V, which are also referred to as right singular vectors, are the eigenvectors of the sentence similarity matrix A^TA. The matrix Σ is an $n \times n$ diagonal matrix whose diagonal elements are non-negative singular values sorted in descending order.

Reduced Dimension Representation

Once the Singular Value Decomposition of the term by sentence matrix is performed, the final step is dimensionality reduction. Because of the dimensionality reduction, noisy relationships are suppressed and important relationships become very clearly visible. As discussed in Chapter 2, there is no straightforward rule that can be used to select the optimal value of dimension r. The decision to keep r singular values or dimensions is done more or less arbitrarily or must be determined experimentally since each data collection is different [24, 54].

In this thesis, dimensionality reduction technique is taken from the work in [13]. The number of dimensions included in the latent space is dependent on the summarization ratio. That is, if the summary needed is p% of the original document, then the top r dimensions are kept in the latent space where r is equal to $(p/100) * n$, n being the number of sentences in the document. Thus, the term by sentence matrix is represented in the latent space by three matrices: U_r, Σ_r and V_r.

The resulting matrix U_r, which is obtained after SVD followed by appropriate dimensionality reduction, groups together terms that co-occur frequently in the document. That is the terms that have high index value in each column of U_r are terms that are very closely related in concept and these terms represent a certain topic in the document. The importance degree of this topic is indicated by the magnitude of the corresponding singular value in the diagonal matrix Σ_r. That is, the first column represents the most important concept or topic; likewise the second column represents the next most important concept or topic, etc. Thus, the matrix U_r can be multiplied by Σ_r to take the significance of each dimension into account. On the other hand, the rows of $V_r * \Sigma_r$ (sentence vectors) represent the semantic representation of sentences in a document and this can be used to compute the similarity between sentences based on their meanings rather than by mere occurrence of terms.

In this thesis, the title of the document is incorporated in the weighted term by sentence matrix, thus the terms of the title directly affect the construction of the latent semantic space. It is also possible to construct a separate vector for the title of the document and project the vector into the latent space of the document. In the context of information retrieval, this is similar to projecting a user query into the latent space of the document collection. However, since projection of the title will be based on the latent structure of the document without the title, terms of the title will have no

effect on the construction of the latent semantic space. This may have a deteriorating effect on the representation of the title.

4.1.3 Sentence Ranking Module

As the primary goal of extractive summarization is to determine the best candidate sentences for summary generation, the sentence ranking module is the core of the summarization system. In this thesis, two new approaches of sentence ranking are proposed. We will refer the proposed approaches as TopicLSA and LSAGraph. These approaches are explained in detail in the coming subsections.

A. TopicLSA

The first approach, TopicLSA, begins with topic identification using the resulting left singular vectors after SVD and dimensionality reduction. The identified topics along with sentence resemblance to title and sentence position are the features used to compute the importance of a sentence for summary generation. Algorithm 4.1 describes how sentence ranking is achieved using TopicLSA. The algorithm has two components: topic identification and sentence score calculation. A detailed description of each component is given below.

Topic Identification

As discussed in the above section, each column of the matrix $U_r*\Sigma_r$ represents salient topics of the document and the values of the matrix represent the importance of each term in the salient topics. The topic or concept identified by the column of the matrix $U_r*\Sigma_r$ can be represented by a set of terms that have high index value in the column. These set of terms consist of one topic term and terms that are related to the topic term. It is our argument that previous LSA-based summarization works can be improved if these set of terms are used to extract semantically important sentences.

That is we assume that by using terms to represent the topics of a document, a wide range of topics in a document can be covered. Hence, a summary generated by extracting sentences that are relevant to the identified topic will have a wide coverage of the document‟s main content. This also provides the possibility to add more than one sentence about an important topic rather than choosing always one sentence for each topic as done in [25]. Hence, unlike previous works which

use the right singular vectors to identify semantically important sentences, we use the left singular vectors to select keywords of the document and use this to compute the importance score of a sentence for summary generation.

Input: the term by concept matrix U, the concept by sentence matrix V^T, the singular matrix Σ, summary compression rate K% and number of sentences in the document L

Output: Importance score or rank of each sentence

1. *Set dimensionality reduction factor r to $\frac{K}{100}$*L*
2. *Compute the matrices U_r *Σ_r and V_r * Σ_r*
3. *For each column of the matrix U_r *Σ_r*
   Select the top m terms that have high index value in the column
4. *Concatenate the selected terms to form a topic vector P*
5. *Remove the first column vector from the matrix $(V_r * \Sigma_r)^T$ to form the title vector T*
6. *Project the topic vector P into latent semantic space.*
7. *For each column vector of the matrix $(V_r * \Sigma_r)^T$ (sentence vector)*
   Sim1 = Cosine similarity (sentence vector, P)
   Sim2 = Cosine similarity (sentence vector, T)
   Pos = 1/sentence position
   Score of sentence = α Sim1 + β Sim2 + γ Pos

Algorithm 4.1: Procedure for summary generation using TopicLSA

The topic identification component of the algorithm selects the top m terms from each column of the reduced matrix U_r*Σ_r. With the aim of selecting sentences that cover a wide range of topics, equal number of terms is selected from each dimension. Since the singular vectors represented by the columns of the reduced matrix Ur*Σ_r are independent of each other, the terms selected from each dimension contain the minimum redundancy.

The number of terms chosen from each dimension is determined by examining the effect of choosing different number of terms on the performance of the summarization method. This will be

explained in Chapter 5. Following the identification of the top terms in each column, a topic vector is constructed by concatenating the selected terms. The topic vector is then projected into the latent space of the original document using the formula $D^{T}U_{r}\Sigma_{r}^{-1}$ where D represents the topic vector. In the context of Information Retrieval, the formula is used to project a user query into latent space [54]. Thus, the topic vector can be considered as a column vector in the reduced topic by sentence matrix and its cosine similarity to other column vectors in the matrix can be computed.

Sentence Score Calculation

Once the topic identification process is completed, the next task is to compute the importance score or rank of each sentence. The significance score of a sentence is computed based on its relevance to the topic vector. Cosine similarity is used to measure the relevance of a sentence to the topic vector. Thus, unlike the method proposed in [13], both long and short sentences have equal chance of being extracted for summary generation. This is because cosine similarity measures the similarity between two sentences based on the unit vectors of the sentences. Moreover, in an attempt to take document genre information into consideration, the position of each sentence in the document and its similarity to the title of the document are used as additional features to compute the significance of a sentence.

News articles in general are written in such a way that the information presented is arranged in descending order of importance. The most important information is placed at the beginning of the news article followed by less important information [56]. Furthermore, the titles of news articles are usually very informative about the content of the article. Thus, we assume that summarization results can further be improved if sentence position and similarity to title are considered when the importance of a sentence is computed. Thus, the significance score of a sentence is computed based on three features: cosine similarity to the topic vector, cosine similarity to the title vector, and sentence position in the document. In order to compute the significance score of a sentence, we define a formula that combines the three features as follows:

$$S(s_i) = \alpha\, Sim1 + \beta\, sim2 + \gamma\, Pos \tag{4.1}$$

where, $S(s_i)$ is the significance score of sentence i,

Sim1 is the cosine similarity of sentence i and the topic vector,

Sim2 is the cosine similarity of sentence i and the title of the document, and
Pos is the position score of sentence i.

The position score of a sentence is calculated using the formula 1/n where n is the position of the sentence in the document. Thus, the first sentence in the document gets the highest score equal to 1 and the last sentence in the document gets the lowest score. The constants α, β *and* γ represent the relative importance of the features and they are determined through experiment.

B. LSAGraph

The second approach, LSAGraph, employs graph-based ranking algorithms to determine the importance score of sentences for summary generation. The resulting sentence vectors after SVD and dimensionality reduction are inputs to LSAGraph. The first task in LSAGraph is to construct a graph where the nodes in the graph are sentence vectors and the edges between the nodes are the cosine similarity between the nodes. Sentences are then ranked according to their importance in the graph which is achieved through the use of graph-based ranking algorithms. In this thesis, we make use of two such algorithms: PageRank and HITS. Each of these activities is discussed in detail below.

Graph Construction

Previous text summarization works based on graph-based ranking algorithms use keyword-based frequency vector to represent each sentence and use various similarity measures to compute similarity between two sentences [46, 47, 48]. However, this way of computing similarity between sentences suffers from the problem of polysemy and synonymy as described in Chapter 2. In this thesis, we attempted to avoid this problem by representing sentences by their corresponding semantic sentence representation described in section 4.1.2. This will allow us to compute sentence similarity on the basis of the topics of a document. This way text summarization is promoted from keyword-level analysis to semantic-level analysis. We use a modified form of equation 2.11 to construct a sentence similarity matrix. The modification is needed because equation 2.11 computes similarity between sentences based on their dot products. The dot product of two sentences is calculated based on the number of word matches between the sentences. However, long sentences tend to have many different terms and this increases the number of word matches between long

sentences and other sentences in the document. To avoid this problem, we use cosine similarity which computes similarity between sentences using the normalized sentence vectors. Normalization is achieved by dividing every sentence vector by its Euclidean length[5].

Hence, we constructed graphs where the nodes in the graphs are represented by the semantic representation of the sentences and the cosine similarity between them establishes the edges between the nodes. For instance, the sentence similarity matrix for the first 10 sentences of a sample text, which is given in annex A, is shown in table 4.2. A value in a particular cell of the matrix represents the similarity of two sentences which is computed based on the semantic representation of the sentences in the reduced latent semantic space.

Table 4.2: Sample sentence similarity matrix

	Sent1	Sent2	Sent3	Sent4	Sent5	Sent6	Sent7	Sent8	Sent9	Sent10
Sent1	1	0.72	0.64	0.79	0.5	0.23	0.44	0.66	0.42	0.57
Sent2	0.72	1	0.28	0.39	0.22	0.12	0.63	0.58	0.08	0.51
Sent3	0.64	0.28	1	0.53	0.46	0.22	0.35	0.11	0.52	0.07
Sent4	0.79	0.39	0.53	1	0.2	0.12	0.13	0.58	0.42	0.5
Sent5	0.5	0.22	0.46	0.2	1	0.05	0.59	0.19	0.28	-0.02
Sent6	0.23	0.12	0.22	0.12	0.05	1	0.48	0.67	0.9	0.76
Sent7	0.44	0.63	0.35	0.13	0.59	0.48	1	0.54	0.5	0.41
Sent8	0.66	0.58	0.11	0.58	0.19	0.67	0.54	1	0.65	0.96
Sent9	0.42	0.08	0.52	0.42	0.28	0.9	0.5	0.65	1	0.66
Sent10	0.57	0.51	0.07	0.5	-0.02	0.76	0.41	0.96	0.66	1

Since similarity between a pair of sentences is computed using cosine similarity, all similarity values lie in the interval [-1, 1]. A pair of sentences that are semantically similar to each other will have a cosine similarity which is close to 1 and a pair of sentences that are semantically different from each other will have a cosine similarity close to -1. For this example, dimensionality reduction is achieved by keeping 20% of the dimensions of the original term by sentence matrix and the term by sentence matrix is weighted using Binary Weight as local weight and Entropy Frequency as global weight.

[5] The Euclidean length of a vector is equal to the square root of the sum of the squares of all elements in the vector.

Sentence Score Calculation

Once a graph is built using the sentence similarity matrix, the next task is to compute an importance score for each node in the graph. These scores are then used to determine the most important sentences (nodes) which will be concatenated to generate a summary. Two graph-based ranking algorithms, PageRank and HITS, are used in this thesis to rank the nodes according to their significance in the graph. Unlike previous summarization works which are based on graph-based ranking algorithms, we modify both PageRank and HITS to take document genre information into account. This is achieved by considering the relevance of a node to the title of a document when the hub or authority value of a node is calculated. This helps to penalize nodes that do not belong to the main topic of the document which is usually represented by the title of the document, particularly in news texts. In the subsequent sections, we discuss the implementations of PageRank and HITS.

PageRank

As discussed in Chapter 2, the original definition of PageRank assumes that hyperlinks are unweighted but the graphs we are considering here are weighted. That is, the edge between two nodes in the graph is assigned with the cosine similarity of the sentences represented by the nodes. Thus, we used the modified PageRank algorithm which integrates edge weights into the graph [46, 48]. Furthermore, we modified PageRank to take the similarity of a sentence to the title of the document into account when the significance score of a sentence is computed. This results in topic-sensetive PageRank similar to the one given in equation (2.7).

In order to implement the modified PageRank algorithm, the mathematical equation of the algorithm given in equation 2.6 is written in its equivalent matrix notation as [48, 57]:

$$p = [dA + (1 - d)B]^T p \tag{4.2}$$

where, p is a vector which represents the PageRank values of the nodes in the graph,

d is a parameter which is typically chosen in the interval [0.1, 0.2], (for this study, d is set to be 0.15)[6],

[6] Various studies have tested different values of d, but generally the value of d is assumed to be around 0.15.

A is a square matrix such that for a given sentence i, all the elements in the i^{th} column are equal to the cosine similarity of the document title to sentence i divided by the summation of the cosine similarity of the title to all sentences in the document, and

the matrix B is a square matrix which is obtained by dividing each element in the sentence similarity matrix by the corresponding row sum.

More specifically, the elements of matrix A are calculated as:

$$A(title, sentence\ i) = \frac{cosine\ similarity(title,\ \ sentence\ i)}{\sum_{sentence\ j \in S} cosine\ similarity(title,\ \ sentence\ j)} \tag{4.3}$$

where S represents the set of all sentences in the document.

Given the sentence similarity matrix C, the elements of matrix B are calculated as:

$$B(i,j) = \frac{C(i,j)}{\sum_k C(i,k)} \tag{4.4}$$

The vector p, that we are looking for in equation 4.2, is the eigenvector of the transpose of the square matrix $D = [dA + (1 - d)B]$ with the corresponding eigenvalue of 1. In order to compute the eigenvector p, we developed an iterative algorithm called power method which is adapted from [48]. The algorithm is shown in algorithm 4.2.

The algorithm starts with setting all values of the eigenvector p with $\frac{1}{N}$. Then, at each iteration, the eigenvector is updated by multiplying with the transpose of the square matrix D and it is normalized by dividing all values in the eigenvector by the largest value in the eigenvector. This is repeated until the column sum of the difference matrix between consecutive eigenvectors is below a threshold. In our experiments, we have observed that the eigenvector converges rapidly. This is because the eigenvector is normalized at each step which limits the number of steps that the algorithm runs for [52].

Input: Square matrix D, convergence threshold θ, matrix size N

Output: eigenvector P

1. $P = (\frac{1}{N}, \frac{1}{N}, \frac{1}{N}, \ldots, \frac{1}{N})^T$
2. $P_0 = P$
3. i = 0
4. for (; ;) {
5. $P = D^T * P_0$
6. $l = \|P\|_{max}$
7. $P = \frac{P}{l}$
8. $\delta = \|P - P_0\|$
9. if ($\delta < \theta$)
10. return P
11. else
12. P_0=P
13. i=i+1 }

Algorithm 4.2: The power method for computing PageRank

For the previous example, the PageRank of each node (sentence) which reflects the sentence‟s importance in the document is shown in bracket next to each node in figure 4.2. A node having a PageRank value close to one is deemed to be very important and a node with PageRank value closer to zero is deemed to be less important. The edges in the graph represent similarity between the sentences based on the sentence similarity matrix given in table 4.2.

HITS

The other graph-based ranking algorithm that we used to rank the nodes or sentences in the similarity graph is HITS. As explained in Chapter 2, the HITS ranking algorithm assigns two scores for each node in the graph, hub and authority score. The hub score is a measure of the outgoing links of a page and the authority score is a measure of the incoming links of a page. However, when the HITS algorithm is applied on an undirected graph, as in this thesis, the hub and authority score of a node are equal.

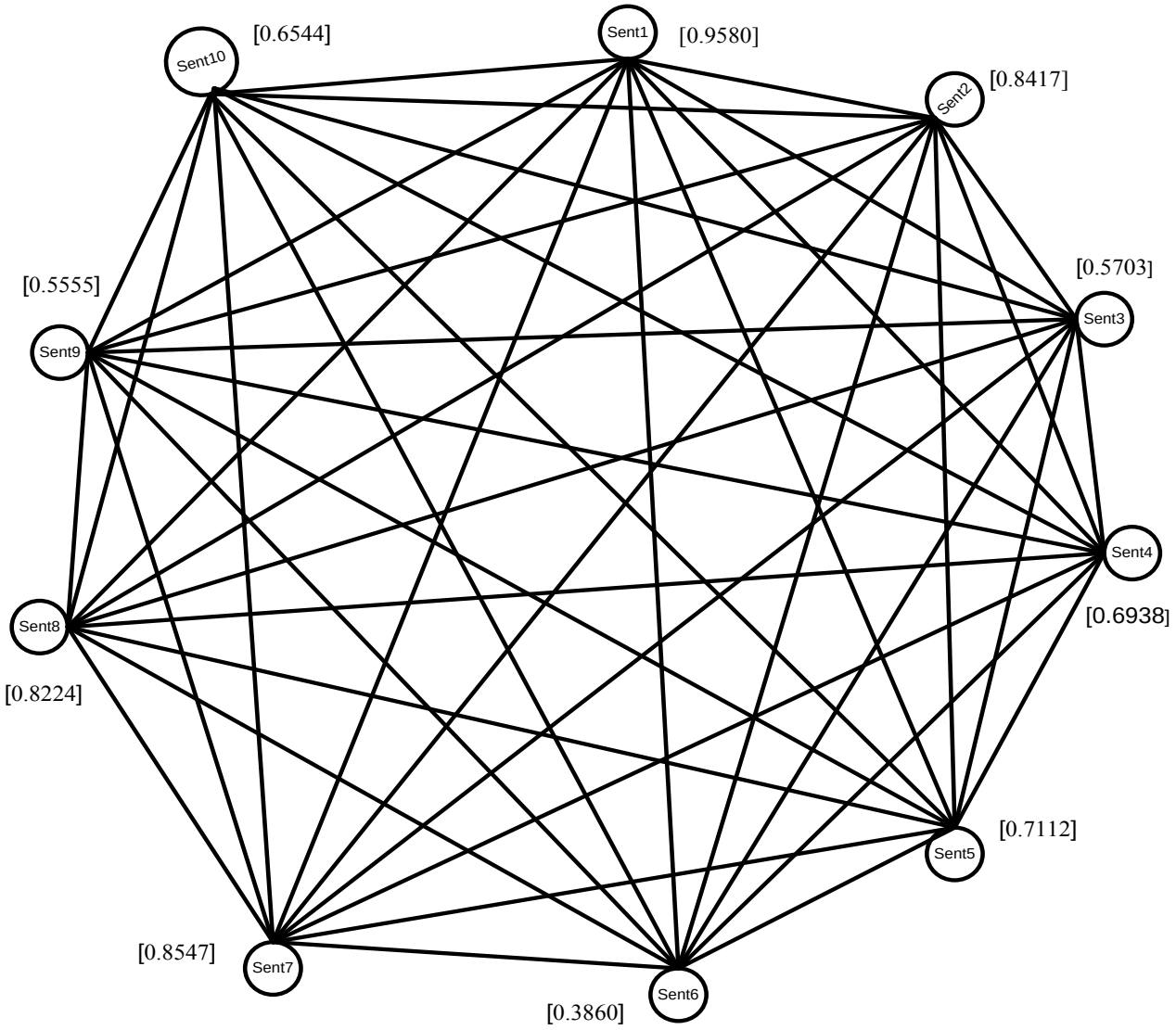


Figure 4.2: Weighted sentence similarity graph for the first ten sentences of a sample text

In this thesis, the formula used to compute hub or authority score is adjusted to take into account connection strength between the nodes of a graph. Hence, the modified form of equation 2.4 which is used to compute authority score becomes:

$$HITS_A(v_i) = \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{kj}} HITS_H(v_j) \quad (4.5)$$

where, $HITS_A(v_i)$ is the "authority" score of a vertex v_i,

$HITS_H(v_j)$ is the "hub" score of vertex v_j,

$In(v_i)$ is the set of vertices that link to vertex v_i ,

$Out(v_j)$ is the set of vertices that vertex v_j link to and w_{ji} represents the connection strength between the vertices v_i and v_j.

As shown in equation 4.5, while computing the authority score for a sentence, the hub scores of the linking sentences are multiplied by the weights of the links. The weights of the links are also normalized by the weight sum of the links.

As described in the preeciding sections, one of our contributions in this thesis is modifying PageRank and HITS such that document genre information is considered while ranking nodes. Similar to the modification we made to PageRank, the HITS algorithm is also modified to consider the relevance of a node to the title of a document when the hub or authority value of a node is calculated.

Thus, equation 4.5 is modified as:

$$HITS_A(v_i) = \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out(v_j)} w_{kj}} * Sim(v_j, title) * HITS_H(v_j) \quad (4.6)$$

where sim(V_j, title) represents the cosine similarity between the title of the document and vertex v_j.

Thus, in order to calculate the authority score of vertex v_i, the relevance of all vertices that links to vertex v_i to the title of the document is multiplied by the hub score of the vertices and by the ratio of the connection strength between the vertices. We could have also added the relevance of a vertex to the title of the document to its hub score instead of multiplying it by its hub score, however, we give more emphasis to title relevance by using multiplication.

For the purpose of implementation, we considered the equivalent matrix notation of equation 4.6 which is given as:

$$A = (B^T B)A \quad (4.7)$$

where A is a one dimensional vector which represents authority score and B is a square matrix whose elements are defined as:

$$B(i,j) = \frac{C(i,j)}{\sum_k C(i,k)} * Sim(i, title) \quad (4.8)$$

where the matrix C is obtained from the sentence similarity matrix by dividing each element of the matrix by the corresponding row sum.

Thus, authority score is an eigenvector of the matrix B^TB with the corresponding eigenvalue of 1. In order to compute this eigenvector, we use the power method shown in algorithm 4.3 [51, 52].

Input: Matrix B, convergence threshold θ

Output: eigenvector A

1. A = $(1,1,1,\ldots1)^T$
2. A_0= A
3. i=0
4. for (; ;) {
5. A = $(B^T * B) * A_0$
6. $l = \|A\|_{max}$
7. A = $\frac{A}{l}$
8. $\delta = \|A - A_0\|$
9. if ($\delta < \theta$)
10. return A
11. else
14. A_0 = A
15. i=i+1 }

Algorithm 4.3: Algorithm to compute authority score

The algorithm starts with a uniform distribution. At each iteration, the eigenvector is updated by multiplying with the matrix (B^TB) and it is normalized by dividing all values in the eigenvector by the largest value in the eigenvector. This is repeated until the column sum of the difference matrix between consecutive eigenvectors is below a threshold. Since the eigenvector is normalized at each step, the eigenvector converges rapidly [52].

4.1.4 Sentence Extraction Module

Once the task of preprocessing, semantic model analysis, and sentence ranking is completed, the final task is to generate a summary. As discussed in the above sections, the proposed summarization system employs two different approaches of sentence ranking. According to the sentence ranking approach selected by the user, the summarizer extracts the top ranking sentences until the desired length of the summary is reached.

Summary length is measured in terms of words rather than sentences. This is because most sentences in news texts are of highly varying length. Hence, measuring summary length in terms of words gives a better approximation of the size of the summary. Once sentences are extracted based on their rank and summary length, a summary is generated by ordering the extracted sentences according to their original position in the source document.

In order to facilitate the summarization process, a simple user interface as shown in figure 4.3 is provided to accept the input document which is to be summarized. The user interface also provides the user with the choice of selecting the summarization method and the summarization rate which is to be used in summarizing the input document. For a sample document given at Annex A, the summary generated at 20% extraction rate using the first approach is given in figure 4.3. A portion of the input document is also presented in the user interface.

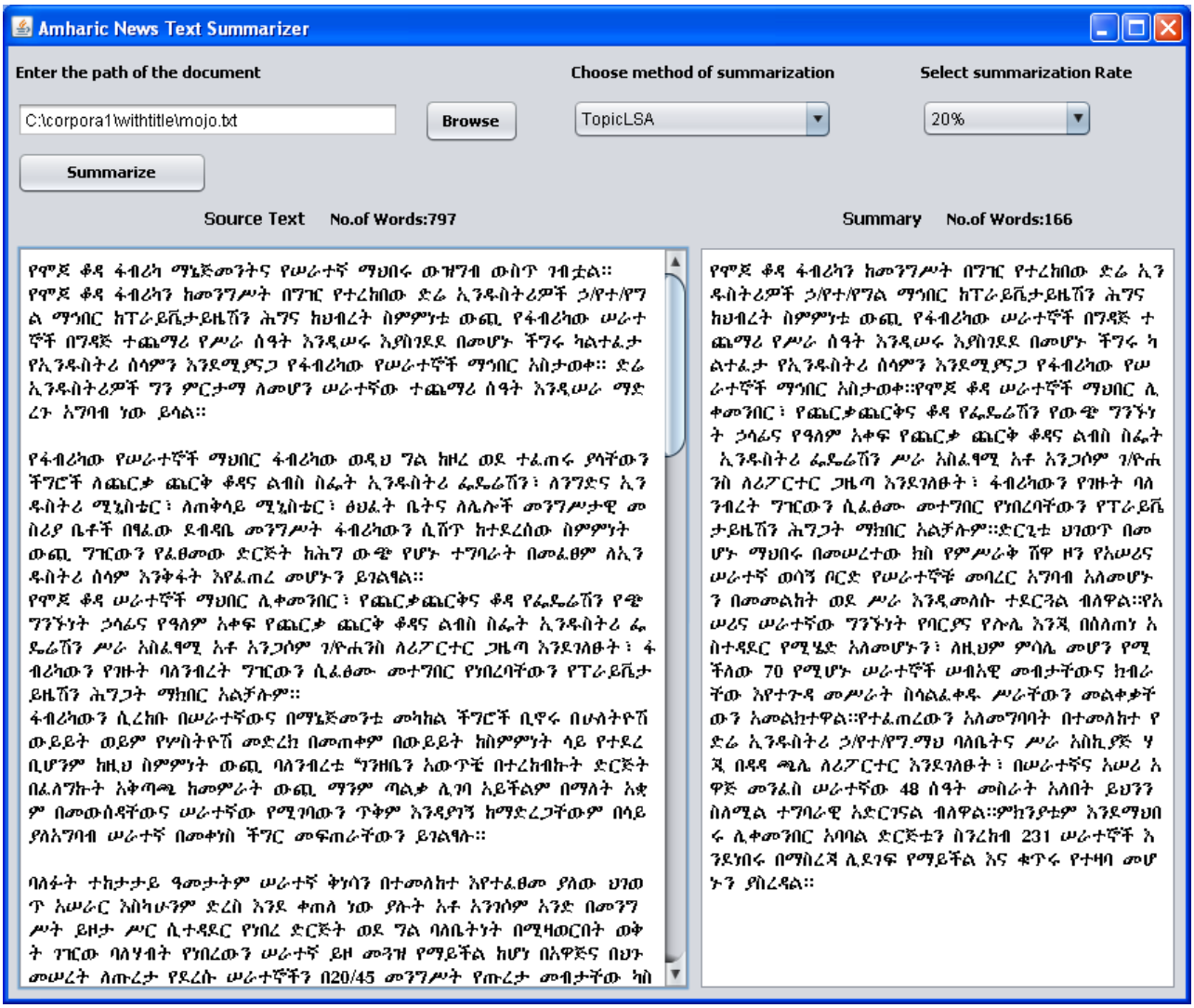


Figure 4.3: Prototype User Interface

4.2 Summary

In this Chapter, we described the main components of the automatic summarizer for Amharic news texts. The summarizer has four main components: document preprocessing, semantic model analysis, sentence ranking, and sentence extraction. The document preprocessing component is accomplished through a series of activities which are adopted from previous studies. The other component of the summarizer, semantic model analysis, represents the input document as a term by sentence matrix and constructs the corresponding term by topic and sentence by topic matrices via singular value decomposition and dimensionality reduction.

Sentence ranking is achieved through two approaches. The first approach, TopicLSA, combines Latent Semantic Analysis with document genre information to select sentences that cover a wide range of topics. The second approach, LSAGraph, runs graph-based ranking algorithms on the semantic sentence similarity graph to select the most important sentences.

CHAPTER FIVE: EXPERIMENT

In this Chapter, we describe the data set, the evaluation metrics, and the results of the summarization system proposed in Chapter four. Though evaluating the performance of the summarization system is an important part of the study, this was not a straightforward task. This is because there is no standard method or well defined criteria for summary evaluation. In the subsequent pages of this thesis, we describe the set of procedures that we used to conduct the experiment.

5.1 Experimental Procedure

In order to evaluate the Amharic news text summarizer, we have carried out the following tasks.

5.1.1 Data Collection

The data set for the experiment consists of 50 Amharic news items whose lengths are in the range of 17 to 44 sentences. These news items were collected from the Amharic version of the Ethiopian Reporter. The reason why the Ethiopian Reporter is selected as a data source is because a large collection of news items on different domains is provided on its Web site. Furthermore, the news items are written in Unicode format which is suitable for our study as we used Java to develop the summarization system. Internally, Java stores characters as 16-bit Unicode characters and to comply with this the document to be summarized should also be represented using the 16-bit representation [41].

News articles with less than 17 sentences were not used in the evaluation due to the fact that summarizing short articles does not make much sense in real applications. In order to evaluate the performance of the summarization system for different domains, the news items we used are from different domains such as sport, technology, politics, etc. Table 5.1 provides the particulars of the evaluation data set.

Table 5.1: Particulars of the evaluation data set

Document Attributes	**Values**
Number of documents	50
Average sentences per document	27
Average words per document	618
Minimum words per document	422
Maximum words per document	1005

5.1.2 Manual Summary Preparation

Six independent evaluators were employed to conduct manual summarization of the 50 news texts contained in the evaluation data set. The first three evaluators prepared manual summaries for 25 news texts and the other three evaluators prepared manual summaries for the remaining 25 news texts. For each news text, each evaluator was requested to rank sentences starting from one for the most relevant sentence up to n for the least relevant sentence, where n is the number of sentences in a given news text. In addition, each evaluator was given a guideline which explains how sentences are ranked. The details of the guideline are provided in Annex B.

Because of the disparities in the evaluator‟s sentence ranking, sentences in each news text are further re-ranked based on their initial ranks assigned by the evaluators. The final rank of a sentence is obtained by taking the average rank of the evaluators. For instance, if a sentence is ranked as 1st, 3rd and 5th by the first, second and third evaluator respectively, then the average rank of this sentence will be 3.

Manual summary for each news text is prepared by extracting the top ranking sentences based on the average rank of the sentences till the required extraction rate is met. Extraction rate is calculated based on the number of words that a document contains. We could also calculate extraction rate based on the number of sentences in a document but the sentences in our data set are of highly varying length. Some sentences are comprised of up to 40 words whereas some sentences are made

of only two words. Thus, calculating extraction rate using the number of sentences does not give better approximation of the size of the summary.

For instance, to prepare the manual summary of a given news text containing 480 words at 30% extraction rate, the top ranking sentences are extracted until the total number of words in the summary reaches 144 (30% of 480). However, in order to produce a summary containing exactly the required number of words, it might be necessary to include only an extract of some sentences. To avoid this, the last sentence to be included in the summary is based on an approximation. For instance, the first three sentences included in the manual summary of the previous news text contain a total of 112 words and the last candidate sentence contains 34 words. Thus, the last sentence will be included in the summary because 30.42% (when the last sentence is included) is a better approximation to 30% than 23.33% (when the last sentence is excluded).

5.2 Performance Evaluation

Evaluation of the summarizer is performed using co-selection measures. As described in Chapter 2, co-selection evaluation metrics assess the quality of the system summary based on the number of sentences that are common to the system summary and the manual summary. The most common co-selection evaluation metrics are recall (R), precision (P), and F-Score (F). The standard definitions of R, P and F are given as [31]:

$$R = \frac{|S_{man} \cap S_{sys}|}{|S_{man}|} \qquad P = \frac{|S_{man} \cap S_{sys}|}{|S_{sys}|} \qquad F = \frac{2RP}{R+P} \tag{5.1}$$

where S_{man} and S_{sys} are the summaries produced manually and by the system respectively.

Since F-Score gives the same importance to precision and recall, using it as an evaluation measure is a good trade-off between precision and recall. Hence, throughout our experiment, F-Score is used to measure the summarization system.

As described in Chapter 4, in this thesis we have proposed two methods of sentence ranking which we reffered as TopicLSA and LSAGraph. Hence, performance evaluation of the Amharic news text summarizer is conducted using the two approaches separately. In the subsequent sections, we discuss evaluation results of the summarizer for the two summarization approaches.

5.2.1 Performance Evaluation of TopicLSA

As described in Chapter 4, TopicLSA summarizes a document based on three features: sentence similarity to the main topic of the document, sentence similarity to title, and sentence position in the document. These features are weighted and linearly combined to obtain the score of each sentence in the document for summary generation. The main topic of the document is found by selecting a set of terms from each column of the reduced term by topic matrix which are then concatenated to construct a topic vector. To have an idea of the relative success of this approach among other LSA based summarization methods, we have compared our method with the summarization methods presented in [13, 25]. But first we conducted several experiments to set the parameters that we used in TopicLSA.

Parameters and Settings

The parameters that we considered to experiment with are the number of terms to use for construction of a topic vector, the different combination of weighting functions that we discussed in Chapter 2, and the weights of the features, α, β *and* γ used in equation 4.1. In an attempt to find the best weights that are to be used in equation 4.1, we performed experiments on a subset of our dataset with different weights constrained between 0 and 1. However, we did not get any considerable improvements. Furthermore, to find a better combination of weights, it is necessary to experiment for several weighting combinations which is usually conducted using machine learning algorithms [60]. Hence, in this thesis we decided to use the same weight, equal to 1, for all features.

In order to investigate the possible effect of the number of terms used to construct a topic vector on the performance of the system, we have run the summarizer by selecting 3, 5, 6, 8, 10, 15, 20, 30, and 50 terms from each column. Furthermore, each selected number of terms is experimented for each combination of weighting function. For convenience of displaying results, F-Score is used as a measure of performance throughout this thesis. Figure 5.1 shows the average F-Score of the 16 weighting functions for each selected number of terms. For all term selections, the manual summary and the system summary are produced at 20% and 30% extraction rate. For each term selections, the average F-Score obtained using the 16 combinations of weighting functions is shown in Figure 5.1.

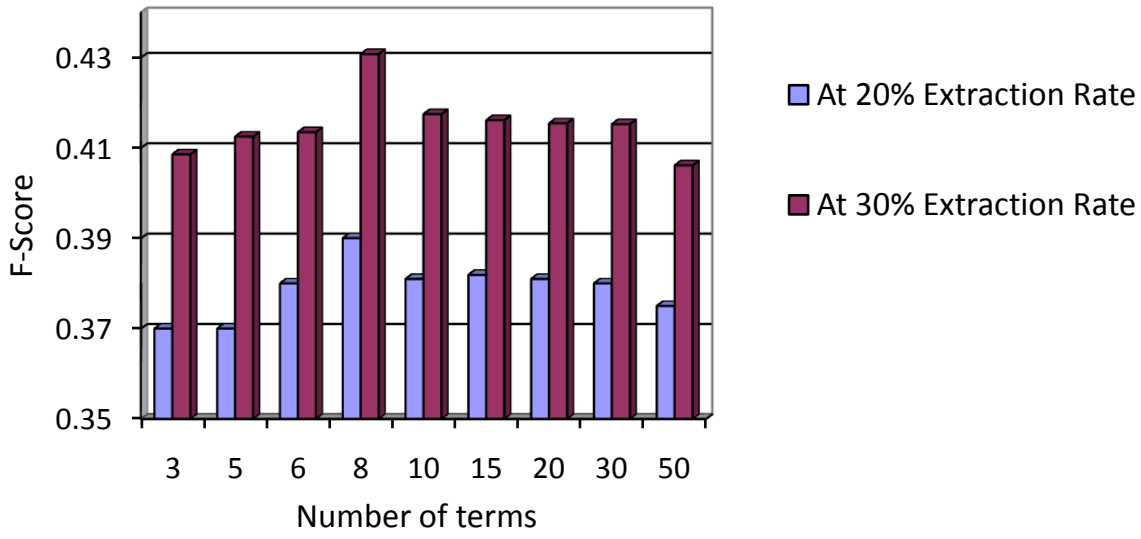
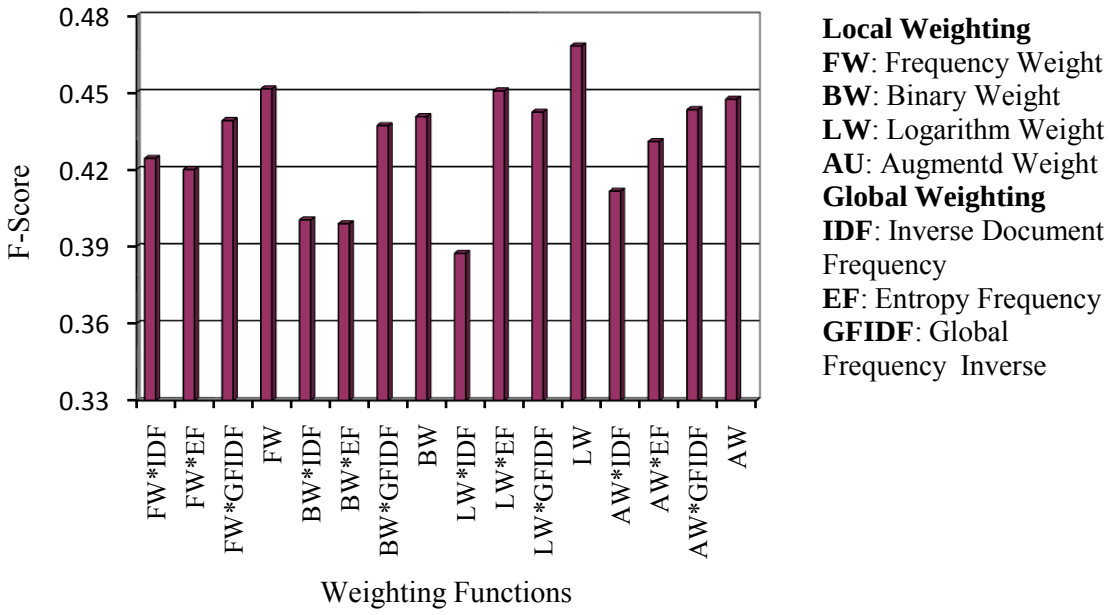
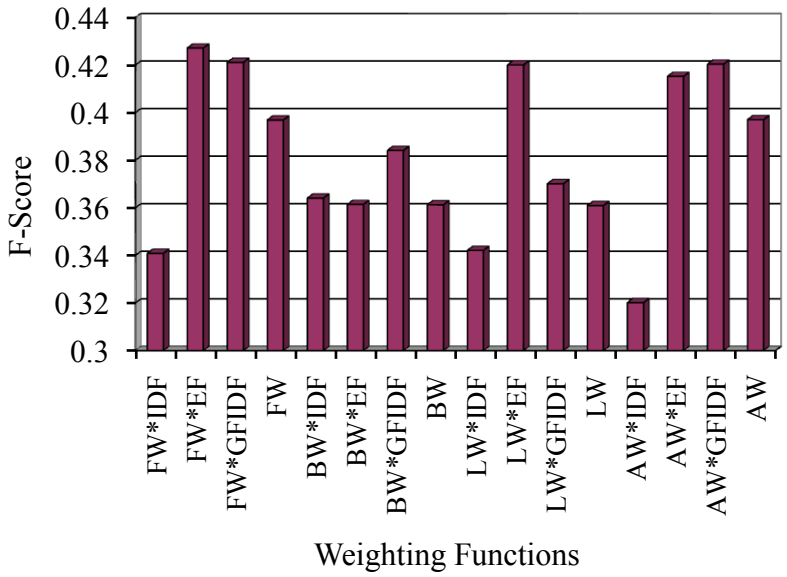


Figure 5.1: Average F-Score of TopicLSA for different selection of terms

As shown in the figure, for both extraction rates, the best average F-Score is obtained when the number of terms selected from each column of the term by topic matrix is 8. As explained in the previous Chapters, each column of the term by topic matrix represents the topics of the document. In our method, these topics are represented by a set of terms which have high index value in a column. As evidenced in figure 5.1, relatively low F-Scores are obtained when the number of selected terms is too small or too large. That is, too small number of terms is not adequate enough to capture the topics of the document and too large number of terms will contain unimportant topic of the document. The influence of the 16 weighting functions on the performance of TopicLSA is shown in figure 5.2. The results shown in the figure are generated when the number of terms selected from each column is 8.



(a)



(b)

*Figure 5.2: The influence of different weighting functions on the performance of TopicLSA (a) at 30% extraction rate and (b) at 20% extraction rate. The meaning of the x-axis labels is as follows: Local Weighting * Global Weighting.*

The figure shows that at 30% extraction rate the best performing weighting function is logarithmic weight coupled with no global weighting function. At 20% extraction rate the best performing weighting function is frequency weight coupled with entropy frequency. Furthermore, it can be observed that at 30% extraction rate combining global weighting functions with local weighting functions has resulted in performance degradation of the summarization method. On the contrary, at 20% extraction rate local weighting functions when used alone have resulted in performance degradation.

The variation in the performance of the weighting funcions at different extraction rates shows that the summarization method behaves differently depending on the number of dimensions retained in the latent space. This is because, in this thesis the number of dimensions kept in the latent space is dependent on the extraction rate.

Comparison of TopicLSA with other LSA-Based Summarization Methods

In this thesis, we have argued that previous LSA-based summarization works can be improved if the topics of the document identified via LSA are first represented by a set of terms. Hence, unlike previous works which use the right singular vectors to identify semantically important sentences, we used the left singular vectors to select keywords of the document which provide a fine-grained representation of the topics of the document. The selected keywords along with genre specific features, sentence similarity to title and sentence position, are used to rank sentences for summary generation.

In order to validate our argument, we have developed the summarization methods presented by Steinberger in [13] and Gong and Liu in [25] for Amharic and compared their F-Scores with that of our method. We will refer these two approaches as baselines throughout the rest of the thesis. There are also other LSA-based summarization methods to which we can compare our method‟s result. However, unlike our method, these methods are developed for multi-document summarization or they are query oriented and hence, we have not compared our method‟s result with them. Apart from adding Amharic preprocessing module to the summarization methods, every aspect of the adopted methods is identical to their original versions.

Comparison of the summarization methods is conducted at 20% and 30% extraction rates. The weighting functions used in our method, based on our result shown in figure 5.2, are frequency weight coupled with entropy frequency and logarithmic weight coupled with no global weighting function for 20% and 30% extraction rates, respectively. The F-Scores of the baselines are investigated for the 16 weighting functions and the maximum scores are taken for comparison. Comparison results are shown in table 5.2.

As evidenced by table 5.2, TopicLSA significantly outperforms the baselines. We attribute our method‟s success to the fact that it employs both LSA and document genre information to select important sentences. More importantly, in our experiments we have observed that the baselines favor long sentences for inclusion in the summary. This has degraded the performance of the summarization methods as in most cases the evaluators have selected sentences that are short but very informative of the topics of the document.

Table 5.2: Comparisons of LSA-based summarization methods

Summarization Method	**F-Score**	
	At 20% Extraction Rate	**At 30% Extraction Rate**
TopicLSA	**0.42**	**0.47**
Steinberger‟s Method	0.23	0.34
Gong & Lui‟s Method	0.26	0.32

This problem was also addressed in Steinberger‟s study and the author has proposed sentence compression algorithms to tackle the problem. However, the author has shown that even with the proposed solutions the system‟s performance has not greatly improved.

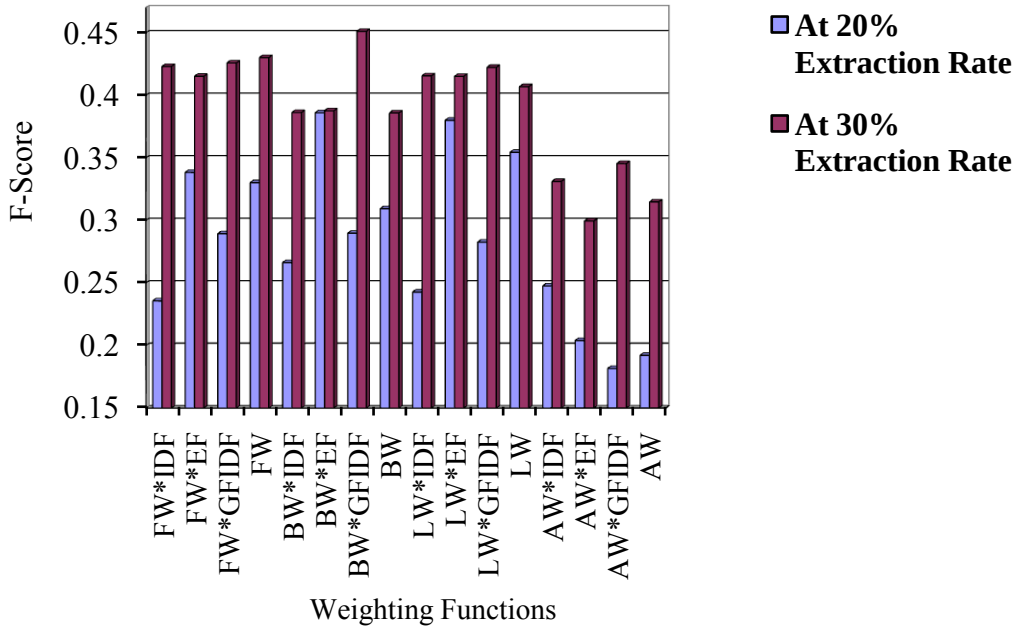
TopicLSA, however, is less prone to the problem of favoring long sentences. This is because two of the three features that we used to rank sentences are based on cosine similarity which normalizes sentences by the number of words in the sentences. Hence, both long and short sentences have equal chance of being included in the summary.

Motivated by the success of TopicLSA, we have also investigated the performance of TopicLSA without using document genre information. This helps us to measure the contribution of document genre information towards the overall success of TopicLSA. To achieve this, we have set the weights of the genre specific features (sentence similarity to title and sentence position) to zero which provides us a version of TopicLSA that does not use document genre information. Evaluation results have shown that the new method has an F-Score of 0.28 and 0.40 at 20% and 30% extraction rates, respectively which is lower than the F-Score obtained using the original TopicLSA but better than the baselines. This shows that document genre information plays an important role to the success of TopicLSA. More importantly, the obtained result supports our argument described at the beginning of this section.

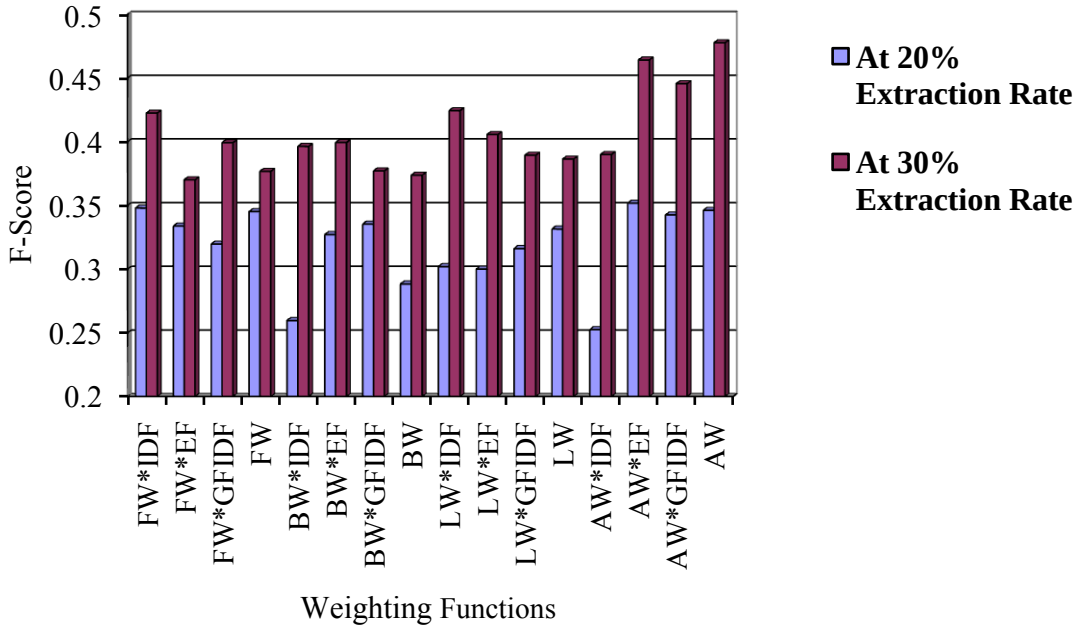
5.2.2 Performance Evaluation of LSAGraph

As described in Chapter 4, LSAGraph is a combination of Latent Semantic Analysis and graph-based ranking algorithms. The main argument here is that the performance of graph-based summarization systems can be improved if the semantic representation of sentences obtained via LSA is used to construct a graph. This is assumed to promote the task of summarization from keyword analysis to semantic analysis. Furthermore, LSAGraph makes use of document genre information to further improve its performance.

In our experiments, two graph-based ranking algorithms were considered: PageRank and HITS. Both ranking algorithms are modified to take the similarity of a sentence to the title of a document into consideration when the rank of a sentence is computed. This is referred to as document genre component of LSAGraph. Evaluation result of LSAGraph using the two graph-based ranking algorithms for each combination of weighting functions is given in figure 5.3. The results shown in figure 5.3 are generated at 20% and 30% extraction rates.



(a)



(b)

Figure 5.3: Performance evaluation of LSAGraph using (a) PageRank and (b) HITS for different combinations of weighting functions

As can be seen in figure 5.3, at 30% extraction rate LSAGraph using PageRank attains the best F-Score when the local weighting function is Binary Weight and the global weighting function is Global Frequency Inverse Document Frequency. At 20% extraction rate, it attains the best F-Score when Binary Weight is coupled with Entropy Frequency. At 30% extraction rate, LSAGraph using HITS attains the best F-Score when Augmented Weight is coupled with no global weighting function. At 20% extraction rate, it attains the best F-Score when Augmented Weight is coupled with Entropy Frequency. As described earlier, these results are obtained when LSAGraph makes use of document genre information.

Comparison of LSAGraph with other Graph-Based Summarization Methods

In order to measure the contribution of using LSA and document genre information in graph-based ranking algorithms, we have also evaluated the performance of PageRank and HITS without the new features: LSA and document genre information. PageRank without the new features is similar to the summarization approach proposed in [48] whereas HITS without the new features is similar to what was proposed in [46]. We refer PageRank and HITS without the new features as baselines. The baselines, like LSAGraph, construct a term by sentence matrix and use it to construct a sentence similarity graph, but the matrix does not undergo Singular Value Decomposition and dimensionality reduction.

The baselines also do not use document genre information. The F-Scores of the baselines are investigated for the 16 weighting functions and the maximum scores are taken for comparison with the baselines. Table 5.3 shows the maximum F-Scores of the baselines and the base system at 20% and 30% extraction rates. The symbols used in the comparisons of the Graph-based summarization methods are as follows:

- LSAGraph + PageRank: LSAGraph using PageRank.
- LSAGraph + HITS: LSAGraph using HITS.
- Baseline1: PageRank without LSA and document genre component.
- Baseline2: HITS without LSA and document genre component.
- LSAGraph + PageRank – Genre: LSAGraph using PageRank and without document genre component.

- LSAGraph + HITS – Genre: LSAGraph using HITS and without document genre component.

Table 5.3: Comparisons of Graph-based Summarization Methods

Summarization Method	F-Score	
	At 20% Extraction Rate	At 30% Extraction Rate
LSAGraph + PageRank	**0.38**	**0.45**
LSAGraph + HITS	**0.35**	**0.47**
Baseline1	0.37	0.44
Baseline2	0.35	0.41
LSAGraph + PageRank - Genre	**0.29**	**0.40**
LSAGraph + HITS - Genre	**0.26**	**0.36**

As evidenced by table 5.3, at both extraction rates the performance of LSAGraph is slightly better than or comparable to the baselines. However, it is not clear if the slight improvement is to be attributed to the use of LSA or document genre information. In order to identify how far LSAGraph can go without using document genre information, we have evaluated its performance without the document genre component. In other words, the new version of LSAGraph still uses LSA but the ranking algorithms assign importance score to sentences with no consideration of the sentences‟ similarity to the title of the document. The F-Scores of the new version of LSAGraph are investigated for the 16 weighting functions and the maximum scores are shown in the last two rows of table 5.3.

As shown in table 5.3, for both ranking algorithms, LSAGraph has attained low F-Scores when document genre information is not used. The obtained values suggest that LSA alone gives no added advantage and in fact, the results obtained are lower than the baselines. This may be attributed to the size of the reduced dimension that we used in our experiments. As described in Chapter 4, the number of dimensions that we kept in the latent space is dependent on the extraction rate. That is, at 20% extraction rate, the percentage of dimensions retained is 20 and at 30% extraction rate, 30% of the dimensions are retained in the latent space. Since dimension in the latent

space refers to the topics of the document, the percentage of dimensions retained in the latent space may have a strong correlation with the performance of the ranking algorithms.

In order to investigate whether increasing the percentage of dimensions retained has an impact on the performance of the ranking algorithms, we computed the F-Scores of LSAGraph for both ranking algorithms by varying the percentage of dimensions retained. Furthermore, in our last experiments LSAGraph do not use document genre information. This is to clearly identify the contribution of LSA to the performance of graph-based ranking algorithms. Figure 5.4 shows the F-Scores of LSAGraph obtained by varying the retained dimensions. For comparison, the F-Scores of the baselines are also depicted in the figure.

In our experiment, we have observed that the influence of the weighting functions on the performance of LSAGraph vary with the change in the percentage of the retained dimensions. Hence, for each percentage of dimensions retained, we have investigated the F-Scores of the 16 weighting functions and used the weighting function which has attained the maximum F-Score. For convenience of displaying evaluation results, figure 5.4 shows comparisons of the ranking algorithms at only 30% extraction rate.

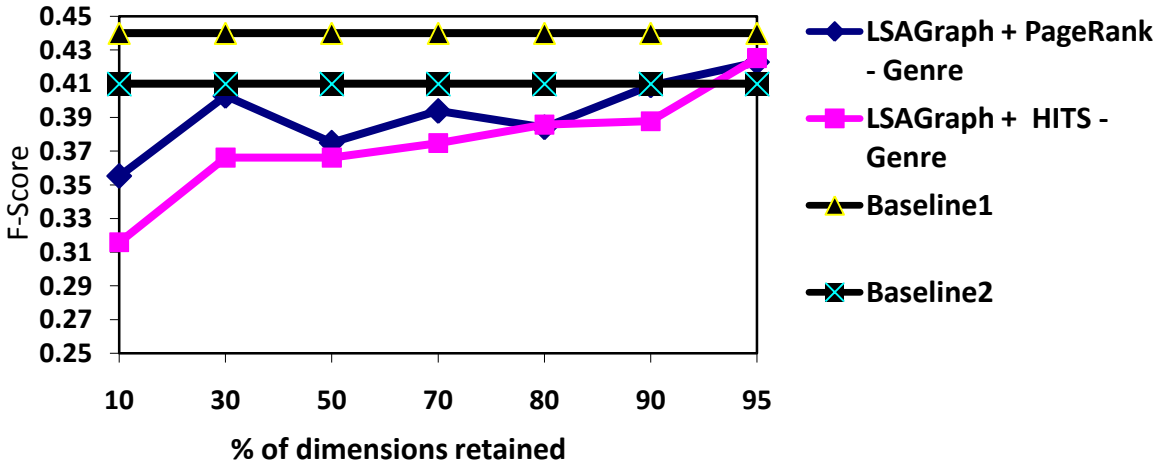


Figure 5.4: Performance of PageRank and HITS with changing LSA dimension

As the figure shows, the performance of LSAGraph without the document genre component improves as the percentage of the retained dimensions increases. That is, it attains comparable F-Score with that of its non-LSA counterparts when more than 90% of the dimensions are retained.

However, using more than 90% of the dimensions in LSA is similar to using the original term by sentence matrix. This shows that dimensionality reduction using LSA gives no added advantage to the graph-based algorithms.

5.3 Discussion

Evaluation of text summarization, particularly generic text summarization, is very challenging. Since neither query nor topic is provided to the summarization task, different evaluators prepare different manual summaries for a given document. In our experiments, we have observed a large degree of differences among the evaluators in assigning importance scores to sentences. This implies that there can be many equally good extracted summaries and this makes evaluation very challenging.

However, performance evaluation of the proposed approaches has shown promising results. Despite the very different mechanisms taken by the proposed approaches to generate a summary, both approaches produced quite comparable performance scores. TopicLSA has significantly outperformed previous LSA based summarization methods. In order to measure the contribution of document genre information towards the overall success of TopicLSA, we have investigated its performance without document genre information. The evaluation has shown that TopicLSA still performs better than the other methods. This supports our argument that the performance of summarization systems using LSA can be improved if the topics of a document identified via LSA are first represented by a set of terms and these are used to identify semantically important sentences. This has helped in providing a fine-grained representation of the topics of a document.

In our second approach, LSAGraph, we have tried to improve the performance of previous summarization works that used graph-based ranking algorithms. In order to achieve this, we have added to existing graph-based summarization systems two new features, LSA and document genre information. Performance evaluation results have shown that these two new features have improved the performance of existing systems. However, we were unable to show performance improvement by using LSA alone. That is, though we used LSA to construct graphs based on the semantic representations of sentences instead of keyword-based sentence frequency vector, this has not brought an improvement in the performance of graph-based summarizers.

Finally, the influence of different weighting functions on the performance of both approaches was investigated and both approaches were observed to be sensitive to the changes of weighting functions. The influence of the weighting functions on the performance of the approaches was also observed to vary with the change in the percentage of the retained dimensions in the latent space.

CHAPTER SIX: CONCLUSION AND RECOMMENDATIONS

6.1 Conclusion

As the amount and availability of textual information increases, techniques to help people obtain and digest the important information in these sources become increasingly important. Today there are numerous documents, papers, reports and articles available in digital form but the information in them is often too abundant to manually search and choose which knowledge one should acquire. Thus, this information must first be automatically filtered and extracted.

The technology of automatic text summarization is critical for dealing with this task. As a result there has been a rapid increase in the number of researches done in the field of automatic text summarization. Many researchers have developed automatic text summarization systems for different languages using different methods. The results of these research works have shown that text summarization can be automated and good results can be obtained. The results of previous automatic text summarization researches for Amharic documents have also shown that summarization for Amharic documents is achievable and very promising.

In this thesis, we proposed two summarization methods based on Latent Semantic Analysis and graph-based ranking algorithms. These methods are incorporated into a single Amharic news text summarization system. The summarization system has three phases: preprocessing, semantic model analysis, and sentence extraction. Preprocessing involves tokenization, normalization, stop-word removal, and stemming and index term selection. The semantic model analysis phase of the system represents the input document as a term by sentence matrix and constructs the corresponding term by topic and sentence by topic matrices via singular value decomposition and dimensionality reduction.

Sentence extraction is achieved through two new approaches proposed in this thesis which we named as TopicLSA and LSAGraph. TopicLSA makes use of the resulting term by topic matrix after SVD and dimensionality reduction to identify the topics of the document. The identified topics along with sentence resemblance to title and sentence position are the features used to compute the importance of a sentence for summary generation. LSAGraph begins with constructing a graph where the nodes in the graph are the resulting sentence vectors after SVD and

dimensionality reduction and the edges between the nodes are the cosine similarity between the nodes. Sentences are then ranked according to their importance in the graph which is achieved through the use of graph-based ranking algorithms.

For experimental evaluations, a dataset consisting of 50 Amharic news texts, which were collected from the Web site of the Amharic version of the Ethiopian Reporter, was prepared. Six evaluators were employed to prepare three manual summaries for each news text contained in the evaluation dataset. Performance evaluations of the two summarization approaches were conducted by comparing the system generated summaries with manual summaries using F-Score which combines the common IR metrics: precision and recall. Despite the very different approaches taken by the proposed methods to generate a summary, both produced quite comparable performance scores.

To have an idea of the relative success of our summarization approaches, we compared the performance of the proposed approaches with previous summarization approaches based on LSA and graph-based ranking algorithms. The comparison was conducted using the same dataset and evaluation results have shown that our summarization approaches have performed better than previous approaches based on LSA and graph-based ranking algorithms. Evaluation also included the study of the influence of different weighting functions on the performance of the proposed summarization approaches. This has revealed the appropriate weighting functions to use with the proposed approaches.

6.2 Contributions of the Thesis

The main contributions of the study are outlined below:

- The general architecture of LSA-based summarizer for Amharic news text is proposed.
- Two new generic text summarization approaches are proposed. The first approach, TopicLSA, uses LSA to identify the topics of a document and combines this with document genre information to generate a summary. The second approach, LSAGraph, combines LSA and document genre information with graph-based ranking algorithms to identify semantically important sentences for summary generation.

- The study has shown the possibility of applying topic-sensitive graph-based ranking algorithms, PageRank and HITS, for generic text summarization which were not considered in previous graph-based summarization works.
- Analysis of different weighing functions on the performance of LSA-based summarization approaches is presented in the study and the appropriate weighting functions to use with LSA are identified.
- In addition, the application of LSA for Amharic text summarization is expected to pave the way for using LSA in other areas of Amharic language processing. This includes information filtering from Amharic documents, classification of Amharic documents, cross-language IR in which Amharic language is one component, and automatic evaluation of Amharic essays.

6.3 Recommendations

The study has shown that summarization can be done automatically for Amharic documents using LSA and graph-based ranking algorithms. However, further research and developmental effort is needed to apply these summarization approaches in a full-fledged automatic summarization system for Amharic documents. Additional features that can be added to increase the performance of the proposed summarization system and future research directions are outlined below:

- Evaluation of automatic text summarization is generally a very difficult task and it was even more difficult in this research as there was no corpus of Amharic documents annotated for summarization. Therefore, it is recommended that such a corpus be prepared for future researches on summarization.
- Evaluation results of the research have shown that LSA is very sensitive to stemming and hence, due effort should be made towards improving the current Amharic stemmer.
- Apart from the preprocessing module, the summarization system developed in this thesis is language independent. Therefore, it will be interesting to evaluate the performance of the system for other languages such as English which enjoy well prepared corpus annotated for summarization.
- Both summarization approaches presented in this thesis take document genre into account in order to improve the quality of the summary. Applying these approaches for other

document genre can be a research direction. Extension of these approaches for multi-document summarization is also another potential direction.

- The first approach, TopicLSA, presented in this thesis computes the significance of a sentence for summary generation based on three features: similarity to the topics of the document, similarity to the title of the document, and sentence position in the document. The weights of these features were determined based on evaluation results of the research. However, the weights of these features can be optimized using machine learning algorithms.
- In this research we have realized that most sentences in news texts are long and such sentences usually contain unimportant clauses. Sentence compression algorithms can be applied to remove unimportant clauses to make the summary more concise and shorter. Therefore, development of a sentence compression algorithm for Amharic can be one research direction.
- The summarization approaches presented in this thesis strive to create a summary with a wide coverage of the document‟s main content. Such summaries and almost all extract based summaries are usually created at the expense of coherence. Hence, different methods such as anaphora resolution, lexical chains, discourse structure, etc. can be used to generate coherent summaries though such methods are only suited for a particular language and demand resources such as a list of discourse cue words and a marked-up training corpus.
- Named entities such as the names of persons, organizations, locations, etc. are the most important elements for the generation of a specific summary. Hence, an effort has been made to incorporate such features into our sentence selection algorithms using a dictionary containing named entities. However, the performance of the system degraded extremely and we opted not to use named entities. The use of named entities for summarization can be considered without affecting system performance using pattern matching or using machine learning techniques. However, pattern matching requires linguists to develop a set of rules for extracting named entities and machine learning techniques require a corpus tagged for named entities. Hence, developing named entity recognizer for Amharic can be a potential research direction and one that is heavily needed.

REFERENCES

[1] Gonenc Ercan, “Automated Text Summarization and Keyphrase Extraction”, Master‟s Thesis, Department of Computer Engineering, Bilkent University, Ankara, Turkey, 2006.

[2] Tristan Miller, “Generating Coherent Extracts of Single Documents Using Latent Semantic Analysis”, Master‟s Thesis, Graduate Department of Computer Science, University of Toronto, 2003.

[3] H.P. Luhn, “The Automatic Creation of Literature Abstracts”, In IBM Journal of Research and Development, pp. 159-165, 1958.

[4] Eduard Hovy, “Text Summarization”, In The Oxford Handbook of Computational Linguistics, pp. 583-598, 2005.

[5] Sparck Jones, Karen, “Automatic Summarising: Factors and Directions”, In Mani and Maybury, eds., Advances in Automatic Text Summarization, 1999.

[6] Lawrence Wong, “ANSES: Automatic News Summarization and Extraction System”, Imperial College, Department of Computing, 1998.

[7] Karel Ježek and Josef Steinberger, “Automatic Text Summarization (The state of the art 2007 and new challenges)”, In Proceedings of Znalosti 2008, pp. 1–12, 2008.

[8] M. Moens, “Automatic Indexing and Abstracting of Document Texts”, In Proceedings of Artificial Intelligence and Law, Voume 8, pp. 343-347, 2000.

[9] Laura Alonso, “Representing Discourse for Automatic Text Summarization Via Shallow NLP Techniques”, PhD Thesis, Departament of Linguistica General, University of Barcelona, Barcelona, Spain, 2005.

[10] John Hutchins, “Summarization: Some Problems and Methods”, In Proceedings of Meaning: The Frontier of Informatics. Informatics 9, pp. 151-173, 1987.

[11] H.P. Edmundson, “New Methods in Automatic Extracting”, Journal of the Association for Computing Machinery, Vol. 16, No. 2, pp. 264-285, 1969.

[12] Gerard Salton, Amit Singhal, Mandar Mitra, and Chris Buckley, “Automatic text structuring and summarization”, In Proceedings of Information Processing and Management, Volume 33, pp. 193 – 207, 1997.

[13] Josef Steinberger, “Text Summarization within the LSA Framework”, PhD Thesis, University of West Bohemia in Pilsen, Czech Republic, 2007.

[14] Eduard Hovy and ChinYew Lin, “Automated Text Summarization in SUMMARIST”, Information Science Institute of the University of Southern California, 1997.

[15] Julian Kupiec, Jan Pedersen, and Francine Chen, “A Trainable Document Summarizer”, Xerox Palo Alto Research Center, 1995.

[16] Chinatsu Aone, Mary E. Okurowski, James Gorlinsky, and Bjornar Larsen, “A Trainable Summarizer with Knowledge Acquired from Robust NLP Techniques”, In Mani and Maybury, eds., Advances in Automatic Text Summarization, 1999.

[17] Regina Barzilay and Michael Elhadad, “Using Lexical Chains for Text Summarization”, In Proceedings of the ACL/EACL‟97 Workshop on Intelligent Scalable Text Summarization, pp. 10–17, 1997.

[18] H. Gregory Silber and Kathleen F. McCoy, “An Efficient Text Summarizer Using Lexical Chains”, Computer and Information Sciences, University of Delaware, 2000.

[19] B. Boguraev and C. Kennedy, “Salience Based Content Characterization of Text Documents”, In Proceedings of the ACL'97/EACL'97 Workshop on Intelligent Scalable Text Summarization, 1997.

[20] Samuel W. K. Chan, Tom B, Y. Lai, W. J. Gao, and Benjamin K. T'sou, “Mining Discourse Markers for Chinese Textual Summarization”, Language Information Sciences Research Centre, City University of Hong Kong, 2000.

[21] W. C. Mann and S. A.Thompson, “Rhetorical Structure Theory: Toward a Functional Theory of Text Organization”, Text, Vol. 8, No. 3, pp. 243-281, 1988.

[22] K. Ono, K. Sumita, and S. Mike, “Abstract Generation Based on Rhetorical Structure Extraction”, Research and Development Center, Toshiba Corporation, 1994.

[23] Tewodros Hailemeskel, “Text Retrieval: An Experiment Using Latent Semantic Indexing (LSI) with Singular Value Decomposition (SVD)”, Master‟s Thesis, Addis Ababa University, 2003.

[24] Thomas K Landauer, Peter W. Foltz, and Darrell Laham, “An Introduction to Latent Semantic Analysis”, Discourse Processes, 25, pp. 259-284, 1998.

[25] Yihong Gong and Xin Liu, “Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis”, In Proceedings of ACM SIGIR, pp. 19-25, 2001.

[26] Gabriel Murray, Steve Renals, and Jean Carletta, “Extractive Summarization of Meeting Recordings”, In Proceedings of Interspeech, Lisboa, Portugal, 2005.

[27] Josef Steinberger and Karel Ježek, “Using Latent Semantic Analysis in Text Summarization and Summary Evaluation”, In Proceedings of the 5th International Conference on Information Systems Implementation and Modelling, pp. 93–100, Czech Republic, 2004.

[28] Jen-Yuan Yeh, Hao-Ren Ke, Wei-Pang Yang, and I-Heng Meng, “Text Summarization Using a Trainable Summarizer and Latent Semantic Analysis”, In Special Issue of Information Processing and Management on An Asian Digital Libraries Perspective, 41(1), pp. 75–95, 2005.

[29] Udo Hahn and Inderjeet Mani, “The Challenges of Automatic Summarization”, In IEEE Computer, volume 33(11), pp. 29–36, 2000.

[30] Luis Perez-Breva and Osamu Yoshimi, “Model Selection in Summary Evaluation”, Massachusetts Institute of Technology, Cambridge, 2002.

[31] Ani Nenkova, “Summarization Evaluation for Text and Speech: Issues and Approaches”, Stanford University, Interspeech 2006 – ICSLP, Ninth International Conference on Spoken Language Processing, Pittsburgh, PA, USA, 2006.

[32] Dragomir R. Radev, Wai Lam, Arda C. elebi, Simone Teufel, John Blitzer, Danyu Liu, Horacio Saggion, Hong Qi, and Elliott Drabek, “Evaluation challenges in large-scale document summarization”, In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, pp. 375-382, July 2003.

[33] Teferi Andargie, “The Application of Machine Learning Technique (NAÏVE BAYES) for Automatic Text summarization (The Case of Amharic News Texts)”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2005.

[34] Helen Adane, “Text Summarization on Amharic Legal Judgments”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2006.

[35] Jagadeesh Jagarlamudi, Prasad Pingali, and Vasudeva Varma, “A relevance-based language modeling approach to DUC 2005”, In Proceedings of the Document Understanding Conference, Vancouver, B.C., Canada, 2005.

[36] Kemal Nuru, “Automatic Amharic News Text Summarization”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2004.

[37] Waleed Al-Sanie, “Towards an Infrastructure For Arabic Text Summarization Using Rhetorical Structure Theory”, Master‟s Thesis, King Saud University, 2005.

[38] Gerard Salton and Michael J. McGill, “Introduction to Modern Information Retrieval”, McGraw-Hill Book Company, 1983.

[39] Meron Sahlemariam, “Concept-Based Automatic Amharic Document Categorization” Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2009.

[40] Yohannes Afework, “Automatic Amharic Document Categorization: the Case of Ethiopian News Agency”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2007.

[41] Tessema Mindaye, “Design and Implementation of Amharic Search Engine”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2007.

[42] Nega Alemayehu and Peter Willett, “Stemming of Amharic Words for Information Retrieval”, University of Sheffield, Sheffield, UK, 2002.

[43] Zelalem Sintayehu, “Automatic Classification of Amharic News Items: the Case of Ethiopian News Agency”, Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2001.

[44] Rada Mihalcea, “Graph-based Algorithms for Information Retrieval and Natural Language Processing”, University of North Texas, 2005.

[45] http://en.wikipedia.org/wiki/Graph_theory, “Graph Theory”, Last accessed on August 29, 2009.

[46] Rada Mihalcea and Paul Tarau, “A Language Independent Algorithm for Single and Multiple Document Summarization” In Proceedings of Second International Joint Conference on Natural Language Processing, pp. 19-24, 2004.

[47] Rada Mihalcea and Paul Tarau, “TextRank: Bringing Order into Texts”, In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2004), pp. 404–411, Barcelona, Spain, 2004.

[48] Gunes Erkan and Dragomir R. Radev, “LexRank: Graph-based Lexical Centrality as Salience in Text Summarization”, Journal of Artificial Intelligence Research 22 (2004), pp.457–479, 2004.

[49] Sergey Brin and Lawrence Page, “The Anatomy of a Large-Scale Hypertextual Web Search Engine”, Computer Networks and ISDN Systems, 30(1–7), pp.107–117, 1998.

[50] http://en.wikipedia.org/wiki/PageRank, “PageRank”, Last accessed on August29, 2009.

[51] J.M. Kleinberg, “Authoritative Sources in a Hyper-linked environment”, Journal of the

ACM, 46(5), pp. 604–632, 1999.

[52] http://en.wikipedia.org/wiki/HITS_algorithm, “HITS Algorithm”, Last accessed on August29, 2009.

[53] T.H. Haveliwala, “Topic-Sensitive PageRank: A Context-Sensitive Ranking Algorithm for Web Search”, IEEETKDE: IEEE Transactions on Knowledge and Data Engineering, 2003.

[54] M.W. Berry, S.T. Dumais, and G.W. O'Brien, “Using Linear Algebra for Intelligent Information Retrieval”, In Proceedings of SIAM Review, Volume 37, No. 4, pp. 573-595, 1995.

[55] Daniel Hailu, “Automatic Amharic Text Summarization”, Master‟s Thesis, Graduate School of Telecommunications & Information Technology, Addis Ababa, Ethiopia, 2006.

[56] Abraham Adefris, “Automatic Multi-Source Amharic News Summarization”, Master‟s Thesis, Graduate School of Telecommunications & Information Technology, Addis Ababa, Ethiopia, 2007.

[57] Jahna Otterbacher, Gunes Erkan, and Dragomir R. Radev, “Using Random Walks for Question-focused Sentence Retrieval”, In Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing, pp. 915 – 922, 2005.

[58] Regis Newo, “Understanding LSI via the Truncated Term-Term Matrix”, Master‟s Thesis, Universität des Saarlandes, Saarbrücken, Saarland, 2005.

[59] Winta Aklok, “Multi-Source Amharic News Summarization Using Machine Learning Approach (Artificial Neural Network)”, Master‟s Thesis, Graduate School of Telecommunications & Information Technology, Addis Ababa, Ethiopia, 2009.

[60] Mohamed Abdel Fattah and Fuji Ren, “Automatic Text Summarization”, In Proceedings of World Academy of Science, Engineering and Technology, Volume 27, pp. 192-195, 2008.

[61] Susan T. Dumais, “Enhancing Performance in Latent Semantic Indexing (LSI) Retrieval”, Technical Report Technical Memorandum, Bellcore, 1992.

[62] Josef Steinberger and Karel Ježek, “SUTLER: Update SummarizER based on Latent Topics”, In Proceedings of TAC'08, NIST, Gaithersburgh, United States, 2008.

[63] Bernard Kolman and David R.Hill, “Elementary Linear Algebra with Applications”, Pearson International Edition, 2008.

ANNEXES

Annex A: Sample News Text and Summaries

የሞጆ ቆዳ ፋብሪካ ማኔጅመንትና የሠራተኛ ማህበሩ ውዝግብ ውስጥ ገብቷል

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ። ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል።

የፋብሪካው የሠራተኞች ማህበር ፋብሪካው ወዲህ ግል ከዞረ ወደ ተፈጠሩ ያላቸውን ችግሮች ለጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን፣ ለንግድና ኢንዱስትሪ ሚኒስቴር፣ ለጠቅላይ ሚኒስቴር፣ ፅህፈት ቤትና ለሌሎች መንግሥታዊ መስሪያ ቤቶች በፃፈው ደብዳቤ መንግሥት ፋብሪካውን ሲሸጥ ከተደረሰው ስምምነት ውጪ ግዢውን የፈፀመው ድርጅት ከሕግ ውጭ የሆኑ ተግባራት በመፈፀም ለኢንዱስትሪ ሰላም እንቅፋት እየፈጠረ መሆኑን ይገልፃል።

የሞጆ ቆዳ ሠራተኞች ማህበር ሊቀመንበር፣ የጨርቃጨርቅና ቆዳ የፌዴሬሽን የውጭ ግንኙነት ኃላፊና የዓለም አቀፍ የጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን ሥራ አስፈፃሚ አቶ አንጋሶም ገ/ዮሐንስ ለሪፖርተር ጋዜጣ እንደገለፁት፣ ፋብሪካውን የገዙት ባለንብረት ግዢውን ሲፈፅሙ መተግበር የነበረባቸውን የፕራይቬታይዜሽን ሕግጋት ማክበር አልቻሉም።

ፋብሪካውን ሲረከቡ በሠራተኛውና በማኔጅመንቱ መካከል ችግሮች ቢኖሩ በሁለትዮሽ ውይይት ወይም የሦስትዮሽ መድረክ በመጠቀም በውይይት ከስምምነት ላይ የተደረሰ ቢሆንም ከዚህ ስምምነት ውጪ ባለንብረቱ “ገንዘቤን አውጥቼ በተረከብኩት ድርጅት በፈለግኩት አቅጣጫ ከመምራት ውጪ ማንም ጣልቃ ሊገባ አይችልም በማለት አቋም በመውሰዳቸውና ሠራተኛው የሚገባውን ጥቅም እንዳያገኝ ከማድረጋቸውም በላይ ያለአግባብ ሠራተኛ በመቀነስ ችግር መፍጠራቸውን ይገልፃሉ።

ባለፉት ተከታታይ ዓመታትም ሠራተኛ ቅነሳን በተመለከተ እየተፈፀመ ያለው ህገወጥ አሠራር እስካሁንም ድረስ እንደ ቀጠለ ነው ያሉት አቶ አንገሶም አንድ በመንግሥት ይዞታ ሥር ሲተዳደር የነበረ ድርጅት ወደ ግል ባለቤትነት በሚዛወርበት ወቅት ገዢው ባለሃብት የነበረውን ሠራተኛ ይዞ መንዝ የማይችል ከሆነ በአዋጅና በህጉ መሠረት ለጡረታ የደረሱ ሠራተኞችን በ20/45 መንግሥት የጡረታ መብታቸው ካስከበረላቸው በኋላ ቀሪውን ሠራተኛ ይዞ መንዝ አለበት። ይህንን ማስፈፀም ሲገባቸው ግን ድርጅቱን በተረከቡ በሦስተኛው ወር ከተረከብኳቸው 231 ሠራተኞች መካከል ከግማሽ በላይ 122 ሠራተኞች የቅነሳ ደብዳቤ ደርሷቸው እንደነበር አስታውቀዋል። ድርጊቱ ህገወጥ በመሆኑ ማህበሩ በመሠረተው ክስ የምሥራቅ ሸዋ ዞን የአሠሪና ሠራተኛ ወሳኝ ቦርድ የሠራተኞቹ መባረር አግባብ አለመሆኑን በመመልከት ወደ ሥራ እንዲመለሱ ተደርጓል ብለዋል።

ለፌዴሬሽኑ በተፃፈው ደብዳቤ ላይ መንግሥት በስሩ የሚተዳደሩ ድርጅቶች ለግል ባለሃብቶች ሲሸጥና የግል ባለሃብቱ ትርፍን ብቻ እንዲያግበሰብስ ሳይሆን በሥሩ ያሉ ሠራተኞችንም እንደማንኛውም ዜጋ ሥርዓቱ በፈቀደው መሠረት እንዲያስተዳድር ሲሆን በሞጆ ቆዳ ፋብሪካ ግን ከዚህ በተለየ መልኩ የሠራተኛውን ሠብአዊ መብትና ክብር የሚነኩ ተግባራት እየተፈፀመ መሆኑን ጠቅሰዋል።

የአሠሪና ሠራተኛው ግንኙነት የባርያና የሎሌ እንጂ በሰለጠነ አስተዳደር የሚሄድ አለመሆኑን፣ ለዚህም ምሳሌ መሆን

የሚችለው 70 የሚሆኑ ሠራተኞች ሠብአዊ መብታቸውና ክብራቸው እየተጐዳ መሥራት ስላልፈቀዱ ሥራቸውን መልቀቃቸውን አመልክተዋል፡፡

አዲስ የህብረት ስምምነት ድርድርንም በተመለከተ ችግር መኖሩን ያመለከቱት አቶ አንገሶም በአዲስ መልክ ተግባራዊ እንዲሆን ስምምነት ላይ የተደረሰበትን የህብረት ሥራ ስምምነት መተግበር እንዳልተቻለ ገልፀዋል፡፡

ሳምንታዊ የሥራ ሰዓት ድልድልም እንዲሁ ከህግ ውጭ እንዲሆን መደረጉን ያስታወሱት አቶ አንገሦም ከዚህ ቀደም ሲሰራበት የነበረው የህብረት ስምምነትም ይሁን በአዲስ መልክ በተዘጋጀው የህብረት ስምምነት የአንድ ሠራተኛ ሳምንታዊ የሥራ ሰዓት 44 ሰዓታት ነው፡፡ በዚሁ ስምምነት መሠረት ሲሰራ ቢቆይም ድርጅቱ በራሱ ፍላጐት በመነሳሳት ለማንም ሳያሳውቅ ሳምንታዊ የሥራ ሰዓት 48 ሰዓት እንዲሆን ማድረጉንና የቅዳሜ የስራ ሰአት ግማሽ ቀን መሆኑ ቀርቶ ሙሉ ቀን እንዲሆን መመሪያ በማውጣቱ የሠራተኛውና ባለንብረቱ ልዩነት የበለጠ እንዲሰፋ ምክንያት እየሆነ መሆኑንም አቶ አንገሦም አስረድተዋል፡፡

ለዓመታት የቆየውን ይህንን ችግር ለመፍታት ማህበሩ ሰፊ ጥረት ቢያደርግም ምንም ዓይነት ምላሽ በማጣቱ የመሰብሰብ መብቱን ተጠቅሞ ባለፈው ሳምንት ስብሰባ ለማድረግና ከኢትዮጵያ ጨርቃጨርቅና ቆዳ ኢንዱስትሪ ፌዴሬሽን ጋላፊዎች ጋር ለመነጋገር ያደረገው ተደጋጋሚ ሙከራ እንኳን ባለመሳካቱና በፋብሪካው ግቢ ውስጥ ለመሰብሰብ ባለመቻሉ ከፋብሪካው ውጭ ሜዳ ላይ ለመሰብሰብ መገደዳቸውንም አመልክተዋል፡፡

የተፈጠረውን አለመግባባት በተመለከተ የድሬ ኢንዱስትሪ ኃ/የተ/የግ.ማህ ባለቤትና ሥራ አስኪያጅ ሃጂ በዳዳ ጫሌ ለሪፖርተር እንደገለፁት፣ በሠራተኛና አሠሪ አዋጅ መንፈስ ሠራተኛው 48 ሰዓት መስራት አለበት ይህንን ስለሚል ተግባራዊ አድርገናል ብለዋል፡፡ የህብረት ስምምነት የሚባለውም በደርግ ጊዜ የተሠራና ለአሁን የማይሆን መሆኑንም ጠቅሰዋል፡፡

በነፃ ገበያ ህግጋት ምርታማ ለመሆን ተጨማሪ ሰዓታትን መስራት አለብን ያሉት ሃጂ በዳዳ ከህግ ውጭ እንዳልሄዱ፤ ይህንኑ ጉዳይ በሚመለከት ለንግድና ኢንዱስትሪ ሚኒስቴርና ለጠቅላይ ሚኒስቴር ፅህፈት ቤት እንደፃፉ አስታውቀዋል፡፡

አቶ አንገሦም ግን የሃጂ በዳዳን ሃሣብ አይቀበሉም፡፡ የህብረት ስምምነታችን 44 ሠዓት የሚል በመሆኑ ይህ መፈፀም አለበት፡፡ ሕጉም የሚለው በህብረት ስምምነቱ ተፈጻሚ እንዲሆን ነው ይላሉ፡፡

በሞጆ ሠራተኞችና ባለንብረት መካከል በተፈጠረው አለመግባባት የሞጆ ከተማ አስተዳደር ጣልቃ ገብቶ የፋብሪካው ባለንብረት የወሰዱት እርምጃ ከህግ ውጭ በመሆኑ በህብረት ስምምነቱ መሠረት እንደቀድሞው ሠራተኛው 44 ሰዓት ብቻ መስራት እንዳለበት ማስታወቁም ተጠቅሷል፡፡

ሃጂ በዳዳ የሠራተኛ ቅነሳን በተመለከተ የተፃፈው ደብዳቤ የተጋነነ ነው ብለውታል፡፡ አምራች ስለሆኑ አዋጭ የሆነውን መንገድ እንዲከተሉ ገልፀዋል፡፡ በሠራተኛው ላይ ደረሰ የተባለውን የሠብዓዊ መብት ጥሰት መሠረት የሌለው ነው ብለዋል፡፡

በሠራተኛ ማህበሩ ሊቀመንር የተፃፈውን ደብዳቤ በመቃወም ድሬ ኢንዱስትሪያል ለፌዴሬሽኑ በፃፈው ደብዳቤ ላይም በሠራተኛ ማህበሩ የተፃፈው ደብዳቤ እውነት የሌለውና ሠራተኛውና ድርጅቱ በሰላም እየሠሩ መሆኑን ይጠቅሳል፡፡ ደብዳቤው ሕገወጥ የሠራተኛ ቅነሳ ተደርጓል ተብሎ የተፃፈው ሃቅነት የሌለው ነው፡፡ ምክንያቱም እንደማህበሩ ሊቀመንበር አባባል ድርጅቱን ስንረከብ 231 ሠራተኞች እንደነበሩ በማስረጃ ሊደገፍ የማይችል እና ቁጥሩ የተዛባ መሆኑን ያስረዳል፡፡

ድርጅቱ ወደ ግል ከዞረ በኋላ ጥቂት ሠራተኞች በራሳቸው ፈቃድ የለቀቁ አሉ የሚለው የድሬ ኢንዱስትሪያል ደብዳቤ የጡረታ እድሜያቸው የደረሱ ሠራተኞችም እንደዚሁ የጡረታ መብታቸውን በማስከበር ድርጅቱን ተሠናብተዋል፣ በሞት የተለዩም አሉ ብሏል፡፡

የሥራ ሠዓትን በተመለከተም በደብዳቤው ላይ እንደተገለፀው በቀን 8 ሰዓት በሳምንት 48 ሰዓት መሥራትና ማሠራት ለድርጅቱ ብቻ ሳይሆን ለመላው የአገሪቱ በአዋጅ የተፈቀደ ስለሆነ ዛሬም ነገም ድርጅቱ በቀን 8 ሰዓት በሳምንት 48 ሰዓት መሠራት አለበት፡፡ ተፈፃሚ የተባለው ግፍና በደል ፈፅሞ የሌለ መሆኑን ድሬ ኢንዱስትሪ በጠቅላይ ሚኒስትር ፅ/ቤትና ለሚመለከታቸው የመንግሥት አካላት ጭምር በፃፈው ደብዳቤ አስታውቋል፡፡

ይህንን የድሬ ኢንዱስትሪያል ደብዳቤ ተከትሎም አቶ አንገሦም እንደገለፁት ደግሞ ችግሩ አሁንም ያለና ባሳለፍናቸው ሳምንትት ሳይቀር ሠራተኛው በግድ ዕረፍት እንዲወጣና የመከፋፈል ሥራም እየተሠራ ነው ይላሉ።

Summaries at 20% Extraction Rate

Summary Produced by TopicLSA

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ።የሞጆ ቆዳ ሠራተኞች ማህበር ሊቀመንበር፣ የጨርቃጨርቅና ቆዳ የፌዴሬሽን የውጭ ግንኙነት ኃላፊና የዓለም አቀፍ የጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን ሥራ አስፈፃሚ አቶ አንጋሶም ገ/ዮሐንስ ለሪፖርተር ጋዜጣ እንደገለፁት፣ ፋብሪካውን የገዙት ባለንብረት ግዢውን ሲፈፅሙ መተግበር የነበረባቸውን የፕራይቬታይዜሽን ሕግጋት ማክበር አልቻሉም።የአሠሪና ሠራተኛው ግንኙነት የባርያና የሎሌ እንጂ በሰለጠነ አስተዳደር የሚሄድ አለመሆኑን፣ ለዚህም ምሳሌ መሆን የሚችለው 70 የሚሆኑ ሠራተኞች ሠብአዊ መብታቸውና ክብራቸው እየተጐዳ መሥራት ስላልፈቀዱ ሥራቸውን መልቀቃቸውን አመልክተዋል።የተፈጠረውን አለመግባባት በተመለከተ የድሬ ኢንዱስትሪ ኃ/የተ/የግ.ማህ ባለቤትና ሥራ አስኪያጅ ሃጂ በዳዳ ጫሌ ለሪፖርተር እንደገለፁት፣ በሠራተኛና አሠሪ አዋጅ መንፈስ ሠራተኛው 48 ሰዓት መስራት አለበት ይህንን ስለሚል ተግባራዊ አድርገናል ብለዋል።ምክንያቱም እንደማህበሩ ሊቀመንበር አባባል ድርጅቱን ስንረከብ 231 ሠራተኞች እንደነበሩ በማስረጃ ሊደገፍ የማይችል እና ቁጥሩ የተዛባ መሆኑን ያስረዳል።

Summary Produced by GraphLSA using PageRank

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ።ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል።ይህንን ማስፈፀም ሲገባቸው ግን ድርጅቱን በተረከቡ በሦስተኛው ወር ከተረከብኳቸው 231 ሠራተኞች መካከል ከግማሽ በላይ 122 ሠራተኞች የቅነሳ ደብዳቤ ደርሷቸው እንደነበር አስታውቀዋል።ድርጊቱ ህገወጥ በመሆኑ ማህበሩ በመሠረተው ክስ የምሥራቅ ሸዋ ዞን የአሠሪና ሠራተኛ ወሳኝ ቦርድ የሠራተኞቹ መባረር አግባብ አለመሆኑን በመመልከት ወደ ሥራ እንዲመለሱ ተደርጓል ብለዋል።በሠራተኛ ማህበሩ ሊቀመንር የተፃፈውን ደብዳቤ በመቃወም ድሬ ኢንዱስትሪያል ለፌዴሬሽኑ በፃፈው ደብዳቤ ላይም በሠራተኛ ማህበሩ የተፃፈው ደብዳቤ እውነት የሌለውና ሠራተኛውና ድርጅቱ በሰላም እየሠሩ መሆኑን ይጠቅሳል።ደብዳቤው ሕገወጥ የሠራተኛ ቅነሳ ተደርጓል ተብሎ የተፃፈው ሃቅነት የሌለው ነው።ምክንያቱም እንደማህበሩ ሊቀመንበር አባባል ድርጅቱን ስንረከብ 231 ሠራተኞች እንደነበሩ በማስረጃ ሊደገፍ የማይችል እና ቁጥሩ የተዛባ መሆኑን ያስረዳል።

Summary Produced by GraphLSA using HITS

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ።የፋብሪካው የሠራተኞች ማህበር ፋብሪካው ወዲህ ግል ከዞረ ወደ ተፈጠሩ ያላቸውን ችግሮች ለጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን፣ ለንግድና ኢንዱስትሪ ሚኒስቴር፣ ለጠቅላይ ሚኒስቴር፣ ዕህፈት ቤትና ለሌሎች መንግሥታዊ መስሪያ ቤቶች በፃፈው ደብዳቤ መንግሥት ፋብሪካውን ሲሸጥ ከተደረሰው ስምምነት ውጪ ግዢውን የፈፀመው ድርጅት ከሕግ ውጭ የሆኑ ተግባራት በመፈፀም ለኢንዱስትሪ ሰላም እንቅፋት እየፈጠረ መሆኑን ይገልፃል።የሞጆ ቆዳ ሠራተኞች ማህበር ሊቀመንበር፣ የጨርቃጨርቅና ቆዳ የፌዴሬሽን የውጭ ግንኙነት ኃላፊና የዓለም አቀፍ የጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን ሥራ አስፈፃሚ አቶ አንጋሶም ገ/ዮሐንስ ለሪፖርተር ጋዜጣ እንደገለፁት፣ ፋብሪካውን የገዙት ባለንብረት ግዢውን ሲፈፅሙ መተግበር የነበረባቸውን የፕራይቬታይዜሽን ሕግጋት ማክበር አልቻሉም።ለዓመታት የቆየውን ይህንን ችግር ለመፍታት ማህበሩ ሰፊ ጥረት ቢያደርግም ምንም

ዓይነት ምላሽ በማጣቱ የመሰብሰብ መብቱን ተጠቅሞ ባለፈው ሳምንት ስብሰባ ለማድረግና ከኢትዮጵያ ጨርቃጨርቅና ቆዳ ኢንዱስትሪ ፌዴሬሽን ኃላፊዎች ጋር ለመነጋገር ያደረገው ተደጋጋሚ ሙከራ እንኳን ባለመሳካቱና በፋብሪካው ግቢ ውስጥ ለመሰብሰብ ባለመቻሉ ከፋብሪካው ውጭ ሜዳ ላይ ለመሰብሰብ መገደዳቸውንም አመልክተዋል፡፡

Summaries at 30% Extraction Rate

Summary Produced by TopicLSA

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ፡፡ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል፡፡የሞጆ ቆዳ ሠራተኞች ማህበር ሊቀመንበር፣ የጨርቃጨርቅና ቆዳ የፌዴሬሽን የውጭ ግንኙነት ኃላፊና የዓለም አቀፍ የጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን ሥራ አስፈፃሚ አቶ አንጋሶም ገ/ዮሐንስ ለሪፖርተር ጋዜጣ እንደገለፁት፣ ፋብሪካውን የገዙት ባለንብረት ግዢውን ሲፈፅሙ መተግበር የነበረባቸውን የፕራይቬታይዜሽን ሕግጋት ማክበር አልቻሉም፡፡ይህንን ማስፈፀም ሲገባቸው ግን ድርጅቱን በተረከቡ በሦስተኛው ወር ከተረከብኳቸው 231 ሠራተኞች መካከል ከግማሽ በላይ 122 ሠራተኞች የቅነሳ ደብዳቤ ደርሷቸው እንደነበር አስታውቀዋል፡፡ድርጊቱ ህገወጥ በመሆኑ ማህበሩ በመሠረተው ክስ የምሥራቅ ሸዋ ዞን የአሠሪና ሠራተኛ ወሳኝ ቦርድ የሠራተኞቹ መባረር አግባብ አለመሆኑን በመመልከት ወደ ሥራ እንዲመለሱ ተደርጓል ብለዋል፡፡የአሠሪና ሠራተኛው ግንኙነት የባርያና የሎሌ እንጂ በሰለጠነ አስተዳደር የሚሄድ አለመሆኑን፣ ለዚህም ምሳሌ መሆን የሚችለው 70 የሚሆኑ ሠራተኞች ሠብአዊ መብታቸውና ክብራቸው እየተጐዳ መሥራት ስላልፈቀዱ ሥራቸውን መልቀቃቸውን አመልክተዋል፡፡ሃጂ በዳዳ የሠራተኛ ቅነሳን በተመለከተ የተፃፈው ደብዳቤ የተጋነነ ነው ብለውታል፡፡በሠራተኛው ላይ ደረሰ የተባለውን የሠብዓዊ መብት ጥሰት መሠረት የሌለው ነው ብለዋል፡፡በሠራተኛ ማህበሩ ሊቀመንር የተፃፈውን ደብዳቤ በመቃወም ድሬ ኢንዱስትሪያል ለፌዴሬሽኑ በፃፈው ደብዳቤ ላይም በሠራተኛ ማህበሩ የተፃፈው ደብዳቤ እውነት የሌለውና ሠራተኛውና ድርጅቱ በሰላም እየሠሩ መሆኑን ይጠቅሳል፡፡ደብዳቤው ሕገወጥ የሠራተኛ ቅነሳ ተደርጓል ተብሎ የተፃፈው ሃቅነት የሌለው ነው፡፡ምክንያቱም እንደማህበሩ ሊቀመንበር አባባል ድርጅቱን ስንረከብ 231 ሠራተኞች እንደነበሩ በማስረጃ ሊደገፍ የማይችል እና ቁጥሩ የተዛባ መሆኑን ያስረዳል፡፡

Summary Produced by GraphLSA using PageRank

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ፡፡ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል፡፡ድርጊቱ ህገወጥ በመሆኑ ማህበሩ በመሠረተው ክስ የምሥራቅ ሸዋ ዞን የአሠሪና ሠራተኛ ወሳኝ ቦርድ የሠራተኞቹ መባረር አግባብ አለመሆኑን በመመልከት ወደ ሥራ እንዲመለሱ ተደርጓል ብለዋል፡፡የአሠሪና ሠራተኛው ግንኙነት የባርያና የሎሌ እንጂ በሰለጠነ አስተዳደር የሚሄድ አለመሆኑን፣ ለዚህም ምሳሌ መሆን የሚችለው 70 የሚሆኑ ሠራተኞች ሠብአዊ መብታቸውና ክብራቸው እየተጐዳ መሥራት ስላልፈቀዱ ሥራቸውን መልቀቃቸውን አመልክተዋል፡፡በሞጆ ሠራተኞችና ባለንብረት መካከል በተፈጠረው አለመግባባት የሞጆ ከተማ አስተዳደር ጣልቃ ገብቶ የፋብሪካው ባለንብረት የወሰዱት እርምጃ ከህግ ውጭ በመሆኑ በህብረት ስምምነቱ መሠረት እንደቀድሞው ሠራተኛው 44 ሰዓት ብቻ መስራት እንዳለበት ማስታወቁም ተጠቅሷል፡፡ሃጂ በዳዳ የሠራተኛ ቅነሳን በተመለከተ የተፃፈው ደብዳቤ የተጋነነ ነው ብለውታል፡፡በሠራተኛው ላይ ደረሰ የተባለውን የሠብዓዊ መብት ጥሰት መሠረት የሌለው ነው ብለዋል፡፡በሠራተኛ ማህበሩ ሊቀመንር የተፃፈውን ደብዳቤ በመቃወም ድሬ ኢንዱስትሪያል ለፌዴሬሽኑ በፃፈው ደብዳቤ ላይም በሠራተኛ ማህበሩ የተፃፈው ደብዳቤ እውነት የሌለውና ሠራተኛውና ድርጅቱ በሰላም እየሠሩ መሆኑን ይጠቅሳል፡፡ደብዳቤው ሕገወጥ የሠራተኛ ቅነሳ ተደርጓል ተብሎ የተፃፈው ሃቅነት የሌለው ነው፡፡ምክንያቱም እንደማህበሩ ሊቀመንበር አባባል ድርጅቱን ስንረከብ 231 ሠራተኞች እንደነበሩ በማስረጃ ሊደገፍ የማይችል እና ቁጥሩ የተዛባ መሆኑን ያስረዳል፡፡ይህንን የድሬ ኢንዱስትሪያል ደብዳቤ ተከትሎም አቶ አንገሦም እንደገለፁት ደግሞ ችግሩ አሁንም ያለና ባሳለፍናቸው ሳምንትት ሳይቀር ሠራተኛው በግድ ዕረፍት እንዲወጣና የመከፋፈል ሥራም እየተሠራ ነው ይላሉ፡፡

Summary Produced by GraphLSAusing HITS

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ።ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል።የፋብሪካው የሠራተኞች ማህበር ፋብሪካው ወዲህ ግል ከዞረ ወደ ተፈጠሩ ያላቸውን ችግሮች ለጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን፣ ለንግድና ኢንዱስትሪ ሚኒስቴር፣ ለጠቅላይ ሚኒስቴር፣ ዕህፈት ቤትና ለሌሎች መንግሥታዊ መስሪያ ቤቶች በፃፈው ደብዳቤ መንግሥት ፋብሪካውን ሲሸጥ ከተደረሰው ስምምነት ውጪ ግዢውን የፈፀመው ድርጅት ከሕግ ውጭ የሆኑ ተግባራት በመፈፀም ለኢንዱስትሪ ሰላም እንቅፋት እየፈጠረ መሆኑን ይገልፃል።የሞጆ ቆዳ ሠራተኞች ማህበር ሊቀመንበር፣ የጨርቃጨርቅና ቆዳ የፌዴሬሽን የውጭ ግንኙነት ኃላፊና የዓለም አቀፍ የጨርቃ ጨርቅ ቆዳና ልብስ ስፌት ኢንዱስትሪ ፌዴሬሽን ሥራ አስፈፃሚ አቶ አንጋሶም ገ/ዮሐንስ ለሪፖርተር ጋዜጣ እንደገለፁት፣ ፋብሪካውን የገዙት ባለንብረት ግዢውን ሲፈፅሙ መተግበር የነበረባቸውን የፕራይቬታይዜሽን ሕግጋት ማክበር አልቻሉም።ለዓመታት የቆየውን ይህንን ችግር ለመፍታት ማህበሩ ሰፊ ጥረት ቢያደርግም ምንም ዓይነት ምላሽ በማጣቱ የመሰብሰብ መብቱን ተጠቅሞ ባለፈው ሳምንት ስብሰባ ለማድረግና ከኢትዮጵያ ጨርቃጨርቅና ቆዳ ኢንዱስትሪ ፌዴሬሽን ኃላፊዎች ጋር ለመነጋገር ያደረገው ተደጋጋሚ ሙከራ እንኳን ባለመሳካቱና በፋብሪካው ግቢ ውስጥ ለመሰብሰብ ባለመቻሉ ከፋብሪካው ውጭ ሜዳ ላይ ለመሰብሰብ መገደዳቸውንም አመልክተዋል።በሞጆ ሠራተኞችና ባለንብረት መካከል በተፈጠረው አለመግባባት የሞጆ ከተማ አስተዳደር ጣልቃ ገብቶ የፋብሪካው ባለንብረት የወሰዱት እርምጃ ከህግ ውጭ በመሆኑ በህብረት ስምምነቱ መሠረት እንደቀድሞው ሠራተኛው 44 ሰዓት ብቻ መስራት እንዳለበት ማስታወቁም ተጠቅሷል።

Manual Summary at 20% Extraction Rate

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ።
ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል።
ባለፉት ተከታታይ ዓመታትም ሠራተኛ ቅነሳን በተመለከተ እየተፈፀመ ያለው ህገወጥ አሠራር እስካሁንም ድረስ እንደ ቀጠለ ነው ያሉት አቶ አንገሶም አንድ በመንግሥት ይዞታ ሥር ሲተዳደር የነበረ ድርጅት ወደ ግል ባለቤትነት በሚዛወርበት ወቅት ገዢው ባለሃብት የነበረውን ሠራተኛ ይዞ መጓዝ የማይችል ከሆነ በአዋጅና በህጉ መሠረት ለጡረታ የደረሱ ሠራተኞችን በ20/45 መንግሥት የጡረታ መብታቸው ካስከበረላቸው በኋላ ቀሪውን ሠራተኛ ይዞ መጓዝ አለበት። ይህንን ማስፈፀም ሲገባቸው ግን ድርጅቱን በተረከቡ በሦስተኛው ወር ከተረከብኳቸው 231 ሠራተኞች መካከል ከግማሽ በላይ 122 ሠራተኞች የቅነሳ ደብዳቤ ደርሷቸው እንደነበር አስታውቀዋል። የተፈጠረውን አለመግባባት በተመለከተ የድሬ ኢንዱስትሪ ኃ/የተ/የግ.ማህ ባለቤትና ሥራ አስኪያጅ ሃጂ በዳዳ ጫሌ ለሪፖርተር እንደገለፁት፣ በሠራተኛና አሠሪ አዋጅ መንፈስ ሠራተኛው 48 ሰዓት መስራት አለበት ይህንን ስለሚል ተግባራዊ አድርገናል ብለዋል።

Manual Summary at 30% Extraction Rate

የሞጆ ቆዳ ፋብሪካን ከመንግሥት በግዢ የተረከበው ድሬ ኢንዱስትሪዎች ኃ/የተ/የግል ማኅበር ከፕራይቬታይዜሽን ሕግና ከህብረት ስምምነቱ ውጪ የፋብሪካው ሠራተኞች በግዳጅ ተጨማሪ የሥራ ሰዓት እንዲሠሩ እያስገደደ በመሆኑ ችግሩ ካልተፈታ የኢንዱስትሪ ሰላምን እንደሚያናጋ የፋብሪካው የሠራተኞች ማኅበር አስታወቀ። ድሬ ኢንዱስትሪዎች ግን ምርታማ ለመሆን ሠራተኛው ተጨማሪ ሰዓት እንዲሠራ ማድረጉ አግባብ ነው ይላል። ባለፉት ተከታታይ ዓመታትም ሠራተኛ ቅነሳን በተመለከተ እየተፈፀመ ያለው ህገወጥ አሠራር እስካሁንም ድረስ እንደ ቀጠለ ነው ያሉት አቶ አንገሶም አንድ በመንግሥት ይዞታ ሥር ሲተዳደር የነበረ ድርጅት ወደ ግል ባለቤትነት በሚዛወርበት

ወቅት ገዢው ባለሃብት የነበረውን ሠራተኛ ይዞ መጓዝ የማይችል ከሆነ በአዋጅና በህጉ መሠረት ለጡረታ የደረሱ ሠራተኞችን በ20/45 መንግሥት የጡረታ መብታቸው ካስከበረላቸው በኋላ ቀሪውን ሠራተኛ ይዞ መጓዝ አለበት፡፡ ይህንን ማስፈፀም ሲገባቸው ግን ድርጅቱን በተረከቡ በሦስተኛው ወር ከተረከብኳቸው 231 ሠራተኞች መካከል ከግማሽ በላይ 122 ሠራተኞች የቅነሳ ደብዳቤ ደርሷቸው እንደነበር አስታውቀዋል፡፡ የተፈጠረውን አለመግባባት በተመለከተ የድሬ ኢንዱስትሪ ኃ/የተ/የግ.ማህ ባለቤትና ሥራ አስኪያጅ ሃጂ በዳዳ ጫሌ ለሪፖርተር እንደገለፁት፣ በሠራተኛና አሠሪ አዋጅ መንፈስ ሠራተኛው 48 ሰዓት መስራት አለበት ይህንን ስለሚል ተግባራዊ አድርገናል ብለዋል፡፡ በነባ ገበያ ህግጋት ምርታማ ለመሆን ተጨማሪ ሰዓታትን መስራት አለብን ያሉት ሃጂ በዳዳ ከህግ ውጭ እንዳልሄዱ፤ ይህንኑ ጉዳይ በሚመለከት ለንግድና ኢንዱስትሪ ሚኒስቴርና ለጠቅላይ ሚኒስቴር ፅህፈት ቤት እንደፃፉ አስታውቀዋል፡፡ በሞጆ ሠራተኞችና ባለንብረት መካከል በተፈጠረው አለመግባባት የሞጆ ከተማ አስተዳደር ጣልቃ ገብቶ የፋብሪካው ባለንብረት የወሰዱት እርምጃ ከህግ ውጭ በመሆኑ በህብረት ስምምነቱ መሠረት እንደቀድሞው ሠራተኛው 44 ሰዓት ብቻ መስራት እንዳለበት ማስታወቁም ተጠቅሷል፡፡ በሠራተኛ ማህበሩ ሊቀመንር የተፃፈውን ደብዳቤ በመቃወም ድሬ ኢንዱስትሪያል ለፌዴሬሽኑ በፃፈው ደብዳቤ ላይም በሠራተኛ ማህበሩ የተፃፈው ደብዳቤ እውነት የሌለውና ሠራተኛውና ድርጅቱ በሰላም እየሠሩ መሆኑን ይጠቅሳል፡፡

Annex B: Guideline for Manual Summary Preparation

The purpose of this research is to develop an automatic (computerized) Amharic news text summarizer. The performance of the summarizer is evaluated by comparing its summaries with ideal or gold standard summaries. Ideal summaries are summaries that are prepared manually. In light of this, you are kindly requested to prepare a manual summary for each news text you are given. To prepare a manual summary, all you have to do is rank the sentences of a text according to their importance in describing the main topics of the text. The most important sentence will be given an importance score equal to one and the least important sentence will be assigned n, where n is the total number of sentences in the document.

When you rank sentences, please try to take the following information into consideration.

1. Sentences should be ranked in such a way that a well-structured and well-organized summary can be formed by combining the sentences together. That is, there should be a coherent flow of information about a topic when we jump from one sentence to another.
2. Sentences should be ranked in such a way that all the main topics of a news text are covered. That is, if a candidate sentence contains the same information as one of the sentences which are already ranked, then it should be ranked less than any sentence that is different from the already ranked sentences.
3. When the ranked sentences are concatenated or combined, it should be easy to identify who or what the pronouns or noun phrases in the ranked sentences are reffering to.
4. Oviously, not all ranked sentences will be used in the summary creation. That is, for each news text, we have intended to prepare two manual summaries by taking 20 % and 30% of the top ranking sentences. Hence, sentences should be ranked in such a way that if 20% or 30% of the ranked sentences are selected for summary creation, then the selected sentences should contain the maximum possible information about the main topis of the news.

Annex C: List of Suffixes and Prefixes

Suffixes

ቹ	ችው	ባቸው
ሥች	ው	ባችሁ
ም	ሁ	ቱ
ች	ኝ	ሽ
ን	ና	ይቱ
ዬ	ለት	ሥቹ
ሥ	ላት	የው
ህ	ላቸው	ኞች
ሽ	ላችሁ	ያል
ዋ	በት	ኛ
ችን	ባት	ቸው

Prefixes

እየ	ለ	እስኪ
ሲ	ይ	እንድ
የ	እንዲ	ከነ
የሚ	ስለ	እን
ከ	እስክ	እነ
በ	እንደ	

Annex D: List of Short Words and their Expanded Forms

ት/ቤት	ትምህርት ቤት
ት/ርት	ትምህርት
ት/ክፍል	ትምህርት ክፍል
ሃ/አለቃ	ሀምሳ አለቃ
ሃ/ስላሴ	ሀይለ ስላሴ
ደ/ዘይት	ደብረ ዘይት
ደ/ታቦር	ደብረ ታቦር
መ/ር	መምህር
መ/ቤት	መስሪያ ቤት
መ/አለቃ	መቶ አለቃ
ክ/ከተማ	ክፍለ ከተማ
ክ/ሀገር	ክፍለ ሀገር
ወ/ር	ወታደር
ወ/ሮ	ወይዘሮ
ወ/ሪት	ወይዘሪት
ወ/ስላሴ	ወልደ ስላሴ
ፍ/ስላሴ	ፍቅረ ስላሴ
ፍ/ቤት	ፍርድ ቤት
ጽ/ቤት	ጽህፈት ቤት
ሲ/ር	ሲስተር
ፕ/ር	ፕሮፌሰር
ጠ/ሚኒስትር	ጠቅላይ ሚኒስትር
ዶ/ር	ዶክተር
ገ/ገዮርጊስ	ገብረ ገዮርጊስ
ቤ/ክርስትያን	ቤተ ክርስትያን
ም/ስራ	ምክትል ስራ
ም/ቤት	ምክር ቤት
ተ/ሃይማኖት	ተክለ ሃይማኖት
ሚ/ር	ሚኒስትር
ኮ/ል	ኮለኔል
ሜ/ጀነራል	ሜጀር ጀነራል
ብ/ጀነራል	ብርጋዴር ጀነራል
ሌ/ኮለኔል	ሌተናል ኮለኔል
ሊ/መንበር	ሊቀ መንበር
አ/አ	አዲስ አበባ
ር/መምህር	ርእሰ መምህር
ፕ/ት	ፕሬዝዳንት
ዓ.ም	አመተ ምህረት
ዓ.ዓ	አዲስ አበባ
ዶ.ር	ዶክተር

Annex E: List of Stop-Words

ሁሉ
ሁሉም
ኋላ
ሁኔታ
ሆነ
ሆኑ
ሆኖም
ሁል
ሁሉንም
ላይ
ሌላ
ሌሎች
ልዩ
መሆኑ
ማለት
ማለቱ
መካከል
የሚገኙ
የሚገኝ
ማድረግ
ማን
ማንም
ሰሞኑን
ሲሆን
ሲል
ሲሉ
ስለ
ቢቢሲ
ቢሆን
ብለዋል
ብቻ
ብዛት
ብዙ
ቦታ
በርካታ
በሰሞኑ
በታች
በኋላ
በኩል
በውስጥ
በጣም
ብቻ
በተለይ
በተመለከተ
በተመሳሳይ
የተለያየ
የተለያዩ
ተባለ
ተገለጸ
ተገልጿል
ተጨማሪ
ተከናውኗል
ችግር
ታች
ትናንት
ነበረች
ነበሩ
ነበረ
ነው
ነይ
ነገር
ነገሮች
ናት
ናቸው
አሁን
አለ
አስታወቀ
አስታውቀዋል
አስታውሰዋል
አስካሁን
አሳሰበ
አሳስበዋል
አስፈላጊ
አስገነዘቡ
አስገንዝበዋል
አብራርተዋል
አበራርተው
አስረድተዋል
አስከ
አባክህ
አባክሽ
አባክዎ
አንድ
አንጻር
እስኪደርስ
እንኳ
እስከ
እዚሁ
እና
እንደ
እንደገለጹት
እንደተገለጸው
እንደተናገሩት
እንደአስረዱት
እንደገና
ወቅት
እንዲሁም
እንጂ
እዚህ
እዚያ
እያንዳንዱ
እያንዳንዳችው
እያንዳንዷ
ከ
ከኋላ
ከላይ
ከመካከል
ከሰሞኑ
ከታች
ከውስጥ
ከጋራ
ከፊት
ወዘተ
ወይም
ወደ
ዋና
ወደፊት
ውስጥ
ውጪ
ያለ
ያሉ
ይገባል
የኋላ
የሰሞኑ
የታች
የውስጥ
የጋራ
ያ
ይታወሳል
ይህ
ደግሞ
ድረስ
ጋራ
ግን
ገልጿል
ገልጸዋል
ግዜ
ጥቂት
ፊት
ደግሞ
ዛሬ
ጋር
ተናግረዋል
የገለጹት
ይገልጻል
ሲሉ
ብለዋል
ስለሆነ
አቶ
ሆኖም
መግለጹን
አመልክተዋል
ይናገራሉ

Declaration

I, the undersigned, declare that this thesis is my original work and has not been presented for a degree in any other university, and that all sources of materials for the thesis have been duly acknowledged.

Melese Tamiru

This thesis has been submitted for examination with my approval as an advisor.

MULUGETA LIBSIE (PhD)

Addis Ababa, Ethiopia

October, 2009