Addis Ababa University

College of Natural Sciences

Department of Computer Science

Efficient Language Independent Text Summarization Using Graph Based Approach

Mattias Gessesse Argaw

A Thesis Submitted to the Department of Computer Science in Partial Fulfillment for the Degree of Master of Science in Computer Science

Addis Ababa, Ethiopia

June 2015

Addis Ababa University

College of Natural Sciences

Departement of Computer Science

Efficient Language Independent Text Summarization Using Graph Based Approach

Mattias Gessesse Argaw

The Examining Committee

Name	Signature	Date
Advisor		
Examiner		
Examiner		

Table of Contents

Abstract

Concise and Informative summary facilitates information consumption, by shortening large documents focusing on important topics in the document. Automatic summary generation can be abstractive or extractive where the abstractive generation tries to rewrite concepts in a document the extractive one summarizes by selecting the most important sentences from the document. Much of the automatic text summarization researches have focused on the extractive summary due to the Natural Language Processing (NLP) requirement for the abstractive summary generation, and the requirement of NLP in generating abstract summaries becomes a deterrent factor for further research in abstractive summary generation as NLP itself is yet another Computer Science field which is in its developmental stage. Interest in automating the process of document summarization began in the late 1950, and through the past six to seven decades a number of researchers have introduced different approaches to automatic text summarization. Generally speaking the extractive summarization techniques can be classified as supervised, which requires training data and unsupervised, which doesn't require training data. A summary can also be generic or genre specific. In this thesis we proposed a graph based automatic text summarization algorithm which is generic and unsupervised. In graph based text summarization sentences in the document are represented as nodes of the graph and similarity between them as edges between the nodes, and the nodes (sentences) will be ranked based on their similarity with other sentences.

Previous works using graph based approach, namely TextRank and LexRank adopt already existing ranking algorithms which are iterative and that were originally designed for ranking web pages by analyzing their citation links. Given the fact that there are some differences between the graphical representations of web references and text documents we proposed a new algorithm which will rank each sentence node of a document graphical representation in a more efficient way than PageRank and generate a better informative summary. Two algorithms called Independent Rank (IR) and Sentence Rank (SR) which will be used in combination to rank each sentence.

The IR rank each sentence based on the degree of the sentence and the weight of its edges, giving us the measure of how important that sentence is in that document assuming that a sentence is important if it shares contents with more sentences. SR follows similar line of thinking with the only difference of considering the importance of those sentences to which it is sharing a content i.e. the IR of the sentence, the main assumption being a sentence is important not only depending the size of its edges and its degree but also based on the importance of those sentences to which it is sharing a content with. And to take advantage of our newly proposed approach we suggested a new similarity measures which simply counts the number of contents shared between two sentences regardless of the size of the sentences and we called this new similarity measure Content Overlap (CO).

Our proposed algorithm reduces the polynomial order iterations (which starts with a minimum of n^4 of iterations) of PageRank represented by the big O notation as $\mathbf{O}(n^c)$, where c depends on the number of sentences n, the weight of edges, and the convergence value selected to a maximum of quadratic order iteration represented by the big O notation as $\mathbf{O}(n^2)$ and generates a better informative summary which is evidenced by the improved Recall Oriented Understudy for Gisting Evaluation (ROUGE) result. Our algorithm reported a ROUGE-1 result of 0.5238 on half of 2002 DUC dataset for stemmed and stop words removed summarization which is an elevated improvement both against the top performing system in DUC 2002, the highest of them reporting 0.4405 ROUGE-1 result and TextRank, that uses a graph based approach, with reported ROUGE-1 result of 0.4229 on the same data set.

Generally graph based sentence ranking algorithms are language independent, and our new algorithm is language independent which we have shown by building a prototype which was experimented on English and Amharic test data.

Keywords: Summarization, Graph Based Sentence Ranking, Node Centrality Measures.

Acknowledgment

First and Foremost I greatly praise and thank Dear and Mighty God for revitalizing my life and teaching me good things. In fact all is from HIM, for HIM, to HIM and By HIM.

Then I thank Dr. Fekade Getahun, for giving a form for my shapeless initial thesis idea and for his insightful and friendly support in the work of this thesis.

Finally I thank my family, who at times put me at the edge of my nerve but who always happen to be watching my back, and to my friends who were there for me in difficult times.

List of Tables

List of Equations

List of Figures

List of Acronyms

AAAI	Association for Advancement of Aritificial Intelligence
ATS	Automatic Text Summarization
CO	Content Overlap
DARPA	Defense Advanced Research Projects Agency
DUC	Document Understanding Conference
GSSST	Graphical Sequential Selection of sentences from all Topic
IR	Independent Rank
LSA	Latent Semantic Analysis
ML	Machine Learning
NGSST	Non-Graphical Sequential Selection of sentences from all Topic
NIST	National Institute of Science and Technology
NLP	Natural Language Processing
ORST	Overall relevance of sentence across the topics
PLSA	Probabilstic Latent Semantic Analysis
PLSI	Probabilistic Latent Semantic Indexing
ROUGE	Recall Oriented Understudy for Gisting Evaluation
SR	Sentence Rank
TFIDF	Term Frequency Inverse Document Frequency
TFISF	Term Frequency Inverse Sentence Frequency
TIDES	Translingual Information Detection, Extraction, and Summarization
TIDES	Translingual Information Detection, Extraction, and Summarization
TREC	Text Retrieval Conference

Chapter 1
Introduction

1.1 Background

Advances in information technology has made the creation and distribution of digital contents easier. As our ability to create and distribute these digital contents in news, blogs, social forums and similar platforms increases, our need to automate the process of finding, identifying, categorizing and condensing those contents is becoming apparent. Such needs have led to the evolution of Computer Science occasioning the birth of new computer science fields such as Natural Language Processing (NLP).

Natural Language Processing (NLP) focuses on automating human language processing tasks such as word and sentence tokenization, document categorization, question answering and document summarization. Automatic document summarization, as a part of automated natural language processing, automates the process of document summarization. Summary of a text, produced by automated systems or humans, is defined as "A text that is produced from one or more texts that contain a significant portion of the information in the original text(s), and that is no longer than half of the original text(s)" [1].

The process of generating automatic summary can be abstractive or extractive. Abstractive summary is a human like summary which is a rewriting of the main idea in the document generated by natural language processing algorithms. The algorithm attempts to find main concepts in the document and expressions that best describe it [2] in a way rewriting the original document with a different set of equivalent expressions. Where as extractive summary attempts to summarize a document by extracting important sentences from the document without any rewriting involved.

The classification of summaries as abstractive or extractive is based on the way the summaries are generated. Additionally, summaries can be categorized in other different categories depending on a given parameter we want to emphasize. Among the different categorizations some include genre-specific vs generic, query-based vs general, single document vs multi document, indicative vs informative.

Genre-specific summarization techniques are specific to a given genre, generic summarization techniques are not genre specific. Whereas query-based summarizations summarize in reponse to a specified user query, for example in response to a user keyword/s typed on search engines, general summaries summarize the entire document without considering any query.Depending on whether we are summarizing a single document or multiple documents on a single topic summaries can be classified as single document summarization or multi document summarization. The summaries generated can be indicative that indicates what the document is about without presenting any content or informative that present content of the document in shortened form [1].

As part of Natural Language Processing (NLP) automating document summarization has been researched since the 60's in 1958 Luhn published an article called "The Automatic Creation of Literature Abstracts" in IBM Journal [3] in which the author uses statistical information derived from the word frequency and distribution to determine the relevant sentences for summarizing the document. And a decade later in 1969 Edmundson followed Luhn by publishing his work "New Methods in Automatic Extracting [4]" in which Edmundson introduced additional components, pragmatic words (cue words); title and heading words; structural indicators (sentence location) other than word frequency and distribution in measuring relevance of sentences.

Such heuristic approaches were employed by the early stage of automatic document summarization in the 60's and 70's [3, 4] . Recent researches introduced advanced approaches such as machine learning algorithms [5], mathematical topic modeling [6] and graphical modeling and analyzing approaches [7].

In this thesis we proposed a new graph based automatic text summarization algorithm for Amharic document which is not restricted by the genre of the document to be summarized, and which doesn't require any manual supervision at any stage of the summarization. In addition to their generic independent and unsupervised features graph based summarization algorithms are language independent, to that end we have experimented both on Amharic and English corpus to show the language independent feature of the proposed algorithm.

1.2 Statement of the Problem

Locally a number of researches have been conducted on automatic Amharic document summarization using supervised and unsupervised learning approaches mainly for single document summarization.The works based on unsupervised or semi-supervised approach [8, 9, 10] mainly use topic modeling algorithm, Latent Semantic Analysis (LSA) and/or Probabilistic Latent Semantic Analysis (PLSA) for modeling topics (a topic being the set of semantically related words in a given document).

The automation level of these works is limited because of the following facts:

1. They are genre specific – due to their consideration of genre specific features of a document like position of a sentence, topic sentence, and similarity of a sentence to the topic sentence either while modeling the topic or when ranking sentences.
2. Inherent limitations of the "unsupervised" topic modeling algorithms i.e. Latent Semantic Analysis (LSA) and Probabilistic Latent Semantic Analysis (PLSA)
    a. In Latent Semantic Analysis determining the number of dimensions retained after the singular value decomposition is an empirical issue [11] which should be done experimentally on a given set of documents. Infact this stage in LSA makes it a supervised activity. Therefore one has to find the optimal dimension for a given set of documents to be summarized in advance and this results in the need for training corpus limiting the algorithms approach in using it for any set of document without advance determination of the optimal dimension to be retained.
    b. In Probablistic Latent Semantic Analysis the number of topics which are the latent variables should be determined a priori [12]. This means that we have to know the number of topics covered in a given document in advance which is possible either through stipulation or experimentation. These days even a simple single news article may discuss different topics, therefore using PLSA for modeling requires that the number of topics covered in a document should be known in advance.

In contrast the use of graph based ranking algorithms in extractive text summarization avoids the limitations (problems) mentioned above. The graph based algorithms can generate summary without the consideration of genre specific features of the text document there by avoiding the manual selection of genres of documents to be summarized and the process require no human supervision.

In graph based text summarization nodes are ranked by the value of their centrality measurement. Former works of automatic text summarization based on graph theory [7, 13] adopt Google's PageRank (which is an Eigenvector centrality measuring algorithm) in ranking nodes of a textual graph, and PageRank has a polynomial order complexity which depends upon the number of sentences, the weigth of similarity between the sentences and the threshold value selected for the convergence. In this thesis we propose a new graph based ranking algorithm which has a maximum of quadratic order complexity that rank sentences in a more efficient way than PageRank.

The proposed algorithm is language independent like the other graph based ranking algorithms. Graph based sentence ranking algorithms are language independent because they analyze the structural relationship among sentences of a document that is created from their lexical content similarity. This new graph based algorithm is proposed for generating generic and unsupervised summarization of documents, the algorithm is tested on Amharic and English dataset, it is tested on the two languages to prove its language independent features and to show its performance

improvement against previous works which used alternative graph based ranking and were tested on English data set.

1.3 Objective

General Objective

The objective of this research is to explore the use of graph based sentence ranking and propose a new graph based summarization approach for generic and unsupervised automatic extractive text summarization.

Specific Objectives

The following specific objectives are identified to realize the general objective:

- Conduct detailed review of literature on issues related to text summarization, graph theories and evaluation of automatic text summarization.
- Propose a new graph based sentence ranking algorithm for extractive text summarization.
- Build a prototype based on the newly proposed graph based sentence ranking algorithm for generating extractive summary.
- Conduct an experiment on the performance of the protype using Amharic and English Dataset.

1.4 Application of Results

The thesis introduces a new graph based node ranking algorithm for automatic text summarization, the algorithm facilitates the generation of extractive text summarization in unsupervised manner and it is not affected by the genre of the document being summarized. In addition to that it is language independent, capable of generating a summary for any given language with the appropriate corresponding preprocessing.

The algorithm also reduces the iterative complexities in ranking individual sentences for inclusion into extractive summaries from the polynomial complexity of PageRank to a maximum of quadratic order complexity, where each sentence is ranked just once. This efficiency in computation is achieved without compromising on the quality of the summary generated, rather improving it.

The thesis also serves as an additional work into the existing Amharic text summarization and generic graph based automatic text summarization researches by introducing additional text summarization algorithm for Automatic Text Summarization.

1.5 Research methodology

1.5.1 Corpus Preparation

Automatic text summarization using a graph based approach is a language independent summarization approach because once relationship among sentences of a text document is represented as a set of nodes and vertices the language feature will be a non-determinant factor in measuring the centrality of each node which represents a given textual unit like a sentence, or a phrase.

The proposed new centrality measurement for extractive summarization is tested on two languages, Amharic and English.

The English corpus is obtained from 2002 Document Understanding Conference (DUC) which is part of a Defense Advanced Research Projects Agency (DARPA) program of United States of America, Translingual Information Detection, Extraction, and Summarization (TIDES), which specifically calls for major advances in summarization technology, both in English and from other languages to English (cross-language summarization).

The year 2002 is chosen to make a parallel comparison of this work with previous graph based text summarization approaches which used 2002 DUC data as their test set.

National Institute of Science and Technology (NIST) produced 60 reference sets with sets defined by different types of criteria as event sets, biographical sets, etc. Each set will have 5 to 15 documents with an average of 10 documents. Each document is at least 10 sentence length but there is no maximum sentence length.

Each document set will be of one of the following four types. There will be an equal number of document sets in each category.

1. Documents about a single natural disaster event and created within at most a seven day window.
2. Documents about a single event in any domain and created within at most a seven day window.
3. Documents about multiple distinct events of a single type (no limit on the time window).
4. Documents that present biographical information mainly about a single individual.

The Amharic corpus is 30 Amharic news articles collected from Ethiopian Reporter News Paper, Ethiopian News Agency, Walta Information Center, Addis Admas News paper which have 15 or more sentences each.

1.5.2 Summary Generation

For each of the individual text in the corpus a set of three summaries will be generated, two ideal summaries and one system generated summary.

The system generated summary will be generated by using the prototype that will be built, the two ideal summaries will be generated by human summarizers.

The DUC set already provides different set of ideal summaries both individually for each document and a multi-document summary per document set. For this work purpose the two generic abstracts (not extract) of each document with a length of approximately 100 words or less provided by DUC will be used as a golden standard to compare against the summary that will be generated by the prototype using ROUGE-N measurement.

For the Amharic corpus, due to the challenging nature of getting participants in generating 100 or less word abstractive summary, each document will be given to two individuals who would selected the first high ranked sentences whose number of words is at least 100, which will be taken as human generated (golden summary) against which the system generated summary will be compared.

1.5.3 Evaluation Technique

The system generated summary is evaluated by n-gram co-occurrence with the set of reference summaries submitted by human summarizers. Among the different algorithms for such evaluation, Recall Oriented Understudy for Gisting Evaluation (ROUGE) package introduced by [14], is used. The package contains four evaluation methods called ROUGE-N, ROUGE-L, ROUGE-S and ROUGE-W.

ROUGE-N also known as N-gram Co-occurrence Statistics is selected for evaluation technique in this work. It is selected for two reasons

a. Because ROUGE-N is the evaluation technique used by previous graph based summarization researches.
b. ROUGE-N evaluation is found to be highly co-related with human evaluations.

1.6 Scope and Limitations of the Study

The preprocessing task i.e. tokenizing sentences into bag of words, removing stop words and stemming content words limit the measure of lexical similarity between sentences which in turn affect the rank of sentences. Therefore the accuracy of the algorithm is limited by the accuracy of the preprocessing tasks.

The three preprocessing tasks are language dependent and require a different set of procedures for each language which limits the direct application of systems built using the algorithm for all languages.

The work is scoped to provide extractive automatic summary of a single document. All genres of a document can be used as input and the summary generated by the prototype will be an informative summary gisting the most important sentences of the document.

The algorithm is claimed to be language independent, but even though the actual algorithm is language independent the preprocessings i.e. removing stop words, stemming is inherently a language dependent task.Therefore, the prototype is built to consider only two languages, Amharic and English.

1.7 Organization of the Rest of the Thesis

The remainder of the thesis is organized into five chapters. Chapter Two is literature review where the conceptual (scientific) aspects of Automatic Text Summarization (ATS) including its historical background, current status and the different works and algorithms of automatic text summarization are presented.It also contains details like features of a document that need to be extracted for text summarization; and the methods for extracting those features. Finally Chapter Two also contains section which discusses the Amharic language briefly with regard to its grammar, parts of speech, and morphology.

Chapter Three reviews related works in automatic text summarization, the related work is scoped to focus on those researches in automatic Amharic summarization and those researches which use graph theoretic approach.

The related works are reviewed with regard to their objectives, problem statement, limitations, contributions and justification in terms of the need for this work in that domain.

In Chapter Four we present the new proposed graph based sentence ranking algorithm. The algorithm's advantage in reducing the polynomial iterative complexity of the previous graph

based sentence ranking algorithms into a maximum of quadratic order iterative complexity is emphasized.

The chapter also presents the design and implementation detail of the prototype based on the proposed algorithm.

Chapter Five contains the experimental setting and the experimental result obtained.

Chapter Six concludes the thesis by summarizing the proposed algorithm, the design, implementation and result obtained and forwards recommendations for future work.

Chapter 2
Literature Review

In this Chapter we present a review of literature on the subject of automatic text summarization. We review automatic summarization interms of its historical background starting from its first inception, and detailing on the different types of summaries that are generated automatically, the techniques of generating automatic text summarization and the different algorithms which employ the different techniques as well are presented.

Finally we reviewed the subject of automatic summary evaluation, focusing on the intrinsic evaluation which evaluates the generated summary for its informative content and literary quality.

2.1 History of Automated Text Summarization

Literatures like [1] and [2] assert that interest in text summarization begin around late 1950; the most significant work being [3] which brought the feasibility of automatic text summarization to the horizon.

The work of Luhn [3] uses word frequency and distribution to compute a relative measure of significance, for individual words and then for sentences.

In 1969 Edmundson [4] brings new concept in addition to the usage of word frequency for sentence evaluation, it adds three more additional methods for evaluating sentence importance:

- Cue Method – The Cue Phrase method is based on the assumption that the relevance of a sentence is based on the presence of certain pragmatic phrases. Phrases like 'the conclusion of this paper' indicates a positive relevance for that sentence, and phrases like 'for example' indicate a negative relevance [15].
- Title Method – This method assumes that if there is a content overlap between the title of the document and a sentence in the document then the sentence's relevance increases.
- Position Method – This is a method which considers the position of sentences as a measure of their relevance for inclusion into summary.

Since the 50's and 60's a tremendous progress has been made in automatic text summarization, and a number of different algorithms and approaches have been used and still a number of new researches continue to be conducted on the subject of automatic text summarization even though much of the researches are on extractive summary due to the computational complexity of abstractive summaries according to [16].

2.2 Types of Summarization

Automatic text summarization can be categorized in different ways, depending on which aspect of summarization is to be emphasized. For example if we consider the input we can classify summarization as:

- Single Document Summarization – where only a single document will be fed to the summarizer.
- Multi Document Summarization – where multiple documents on similar topic will be fed to the summarizer.
- Multi Lingual Summarization – where documents written in different language are processed to generate a single language summary.

On the other hand depending on the method used to build the summary the summary can be classified as:

- Extractive summary – where by sentences are evaluated based on their centrality in representing the documents central idea and sentences which are ranked higher will be candidates for inclusion into a summary.
- Abstractive summary – this is human like summary, in which instead of selecting important sentences as they are in the document to be summarized, the summary of abstractive summary will be a rewritten shortened form of the main concepts in the document, paraphrased as per the summarizer wording. Even though abstractive summary has a high potential for condensed summary, the programming required to realize abstractive summary is relatively difficult because it requires Natural Language Processing which in itself is a field under progress [17].

Based on the requirement of training data to be used by the summary generator, summarization can be divided as supervised or unsupervised [17, 16] :

- Supervised – supervised summarization, as the name implies, requires the supervision of the trainer in training the system on specific domains with large training data [17] to learn key features of the document which will later be used for summary generation.
- Unsupervised – contrary to the supervised summarization, unsupervised summarization will not require a training corpus nor other manual interventions to generate a summary. Unsupervised summarization approach as the problem from a different angle by

abandoning the attempt "to learn explicit features that characterize key phrases" from the training data, to exploiting the structure of the text itself (without other corpus) to determine key phrases that appear central to the text [17].

2.3 Document Features for Extractive Summarization Techniques

Gupta and Lehal [2] discussed textual unit's features that are considered while extracting sentences and the methods that are used for extracting those features. In total they have indicated 12 different kinds of features that are mostly used to generate extractive text summarization:

1. **Content Word (Keyword) Features** – content words are linguistic terms which refer to words "such as nouns, most verbs, adjectives, and adverbs that refer to some object, action, or characteristic" [18]. After the content words are identified they will be used as a criteria to select sentences to be included in the summary.

2. **Title Word Feature** – with title word feature sentences which contain words that exist in the title will be considered for inclusion into the summary.

3. **Sentence Location Feature** – with this feature position of sentences will have a certain value based on their position in a document, for example, sentences that are the first in the paragraph and last in the last paragraph to be of high value.

4. **Sentence Length Feature** – with this method, the length of sentences will be considered as a sentence relevance criteria, for example, excluding sentences that are very short or very long from the summary.

5. **Proper Noun Features** – proper nouns are given names for person, places and concepts, ideas, etc and sentences containing proper nouns will have an increased probability to be included in the summary.

6. **Upper Case word feature** – Casing in a literature conveys additional meta data hence sentences containing acronyms or proper names are included in the summary.

7. **Cue Phrase Feature** – cue phrase featuring is one of the earliest techniques in text summarization, with summarizers which consider cue phrases sentences containing cue phrases like However, In conclusion, in contrast are most likely to be included in the summary.

8. **Biased Word Feature** – this technique compares words in sentences against a previously defined biased word list, and if the sentence contains any of the biased words it will be considered for inclusion in summary.

9. **Pronouns Feature** – with this technique sentences which contain pronouns are excluded from list of sentences for inclusion into summary in order to avoid the problem of broken anaphoric references.

10. **Sentence to Sentence Cohesion** – the technique rates sentences based on their content overlap with other sentences of the document, and the sentence which is more overlapped will be ranked higher for inclusion in the summary.

11. **Occurrence of Non-Essential Words** – some words indicate the information is not essential either because the information following those keywords is elaborative, additional or similar. Example of such words are like "because", "furthermore". With this technique sentences which are annotated by such words will be excluded because they are assumed to be redundant.

12. **Discourse Analysis** – a discourse is defined by [19] as an instance of a language use whose type can be classified based on factors such as
    a. Grammatical and lexical choices
    b. And their distribution in
        i. Main versus supportive materials
        ii. Theme
        iii. Style, and
        iv. The framework of knowledge and expectation within which the addressee interpret the discourse.

    Gupta and Lehal [2] suggest that the analysis of a discourse is one of the good features for text summarization to produce coherent, fluent summary and to determine the flow of the author's argument.

    In general the features that are considered for ranking sentences in extractive sumamrization can be either statistical or linguistic.

    The different extractive summary generation methods introduce different solution on how to best extract those features and use those features in selecting/ranking sentences for composing the summary.

In Section 2.4 we present the different methods that are used for extracting document features for extractive summary generation.

2.4 Document Feature Extraction Methods

The main task in most of the automatic extractive document summarization approaches is the identification of important features of a given textual unit of a document which will be used to select those units to construct the extractive summary. The textual units can be words, phrases or sentences. Once the determinant features on the preferred unit of a text (words, phrases, sentences, etc) are identified then those textual units which contain the important features will be selected to construct our extractive summary.

There are a number of feature extractive methods. Here, we present some of the feature extractive methods which we deemed important and pertinent to our work.

2.4.1 A Term Frequency Inverse Document Frequency (tf*idf) based Methods

According to [20] tf*idf is a means of measuring important terms in a document based on the frequency of a word in a document or collection of documents.

As stated previously, one of the features in a document is the ***list of content words***. One way of selecting content words from a document is by using their term frequency (TF), their term frequency being the total number of times a given word appears in the document divided by the total number of words in the document. Yet prioritizing words of a document based on their term frequency is biased by the frequency of non content bearing words, or what are called functional words which are there to form grammatical relationship among words, and which tend to have an exaggerated frequency because they appear in all documents.

For example if we happen to be looking for documents containing the words **"the brown cow"**, and if we simply count the term frequency of each of the words "the", "brown" and "cow" and prioritize the documents based on their containing those words, the word "the" will give us an exaggerated consideration because it is not a content word and it occurs much greater than the other two words. So the term frequency i.e. the number of times a word occurs in a document divided by the total number of words in a document is not a good measure, as this statistics gets higher for words which are not content bearing and appear in almost all documents. Therefore it should be countered by another factor to get a good statistics on content bearing words.

This exaggerated frequency value of non content bearing words is countered by a method called the inverse document frequency or IDF to weigh down the frequency of non-content bearing word, there by pushing up the value of content bearing words.

The inverse document frequency (IDF) is the natural logarithm of the quotient of the total number of documents divided by the number of doucments containing the term. The IDF will be smaller for those words which tend to appear in every document, and the multiplication of the term frequency (TF) by the IDF will result in a new value which is higher for content bearing words than for non content bearing words.

This new measure is known as tf*idf and it is formalized as:

$$tf * idf = tf \left(\log_e \frac{Total\ Number\ of\ Documents}{Number\ of\ Documents\ with\ t\ in\ it}\right) \quad (1)$$

TF*IDF is originally a method used in Information Retrieval to evaluate document relevance with respect to query words, and Text Summarization has adapted that approach to generate automatic summary but as there might not be query words in the case of text summarization, after measuring the tf*idf in the documents the summarizing algorithm considers those nonstop words with higher tf*idf in the document [2] as query words and instead of the documents, sentences will be considered when applied on single document and equation 1 can be modified accordingly as follows:

$$tf * isf = tf \left(\log_e \frac{Total\ Number\ of\ Sentences}{Number\ of\ Sentences\ with\ t\ in\ it}\right) \quad (2)$$

2.4.2 LSA based Methods

The TF*IDF method fails to consider the semantic relationship between words for example even though there are semantic relationships between the term money and wealth, the traditional term frequency models, TF or the TF*IDF, will not consider their semantic relationship and just count the words individually and compute their frequency regardless of their semantic relationship.

One of the methods to solve this problem of failing to account for semantic relationship such as polysemy and synonym is Latent Semantic Analysis. The basic notion of LSA is that the underlying (latent) semantic relationship between words of a document can be learned from the

aggregate of all the word contexts in which a given word does and does not appear, as these contexts provide a set of mutual constraints that largely determines the similarity of meaning of words and sets of words to each other [11].

The first step in LSA is to represent the document as word-to-sentence co-occurrence matrix, where sentences will be columns of the matrix and words rows of the matrix. The entries of the matrix will be the number of times that word appear in that sentence, which is represented by the column.

The next step after forming the word-to-sentence matrix is the use of a mathematical matrix factoring called Singular Value Decomposition (SVD). SVD decomposes a rectangular matrix into the product of three other matrices such that when the three component matrices are matrix-multipied, the original matrix is reconstructed.

The final step in processing the word-to-sentence matrix is the dimension reduction which is done by deleting coeffients in the diagonal matrix starting from the smallest one, the construction with a reduced dimension will create a new co-occurrence matrix where semantically related words will have close values.

Extractive summarization algorithms which use LSA use the list of semantically words induced by the LSA as a criteria for ranking sentenece to be considered in their extractive summary.

2.4.3 Machine Learning Methods

The previous two methods are examples of what is commonly known as centroid based in the context of automatic text summarization. They are named centroid based because they attempt to find the central topic covered in a document as a bag of words and use that "central topic" as a means of measuring sentence's relevance to be included in an extractive summary.

Machine learning methods differ from the previous methods as they essentially models summarization as a classification problem and this involves statistically learning the probability of sentences to be included or not included in a summary from a training document and their extractive summary using machine learning rule [2].

Given a set of training documents and their extractive summaries, sentences are classified as summary sentences and non-summary sentences based on the features that they possess [2].

Different Algorithms have been used in machine learning including the Baye's rule. Neto, Freitas, and Keistner [21] has used Naïve Baye's and C4.5 Machine learning algorithms to learn features of sentences that need to be included in the summary.

Neto, Freitas, and Keistner [21] identified a set of 13 feautres which includes Mean TS-ISF, Sentence Length, Sentence Position, Similarity to Title, Similarity to Keywords, Sentence-to-Sentence Cohesion, Sentence-to-Centroid Cohesion, Depth in the tree, Indicator of main concepts, Occurrence of proper names, Occurrence of anaphors, and Occurrenc of non-essential information and used Baye's rule.

Once these features of a sentence are computed the summary generation consists of the following process.

1. Standard preprocessing is done on the document which are stop-word removal and stemming.
2. Each sentence is converted into its vectorial representation.
3. The features mentioned above are computed, and those feature which are continuous are discretized.

Then two ML algorithms, namely, Naïve Baye's and C4.5 are used to train the machine on the features and a test data is used to generate the summary.

2.4.4 Graph based Methods

Graph based algorithms unlike the preceeding methods, topic modeling and machine learning, doesn't require an identification of a central topic as in the case of topic modelers nor the training corpus with its corresponsing summary set but instead identify important sentences based on the graphical relationship that will be formed from the lexical similarity among sentences of a document.

Graph based methods have been used both in abstractive [22] and extractive summarizations.The basic notion behind graph based algorithms in extractive summary generation is that by representing the document as graph and analyzing it, it is possible to identify issues or topics addressed in the document [2].

Using Graph based methods a document can be modeled as a graph by making document units (which can be words or sentences) nodes of the graph and the level of similarity between the document units as their edge. Once the document is changed into a graph with nodes and weighted edges, one can extract issues or topics addressed in the document by using different graph theories such as degree and centrality measurements.

Representing a document as a graph has two important benefits:

i. Once the document is formed as a graph clusters will be created which allows the choice of coverage for generating summary for query based summaries.Whereas for generic summaries representative sentences from each cluster (sub graph) can be selected.

ii. The second benefit of representing a document in a graph is for the identification of important sentences in the document. Sentences that are represented by the higher cardinality (number of edges connected to that node) will be considered important sentences for inclusion in the summary generation. shows a graphical representation of an imaginary document, nodes with encircled sign representing informative sentences of the document; they are informative because they share similarity with more other sentences in the document

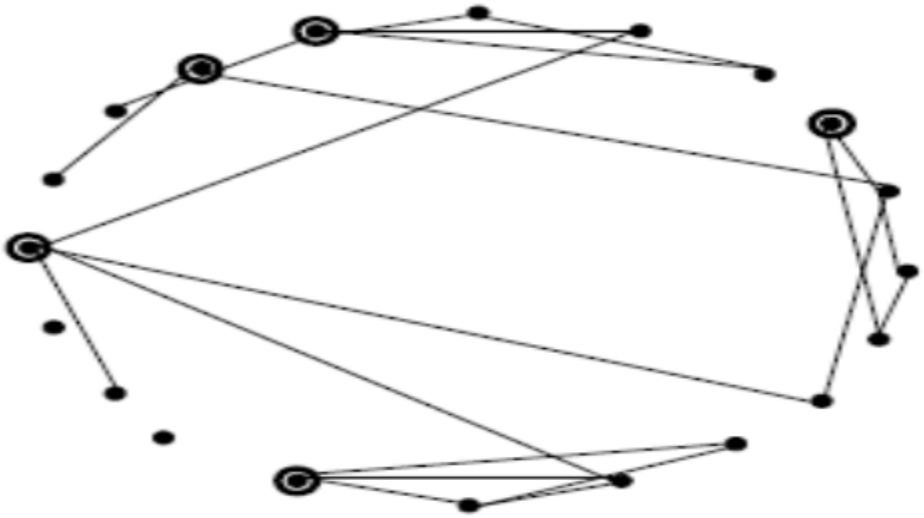


Figure 2-1 : Document Graph *[2]*

Figure 2-1 represents a hypothetical document, where we can see four distinct clusters representing a set of highly related sentences and sentences represented by larger double circles being the dominant sentences in that document because they are related to more sentences.

2.4.5 Hybrid Methods

Hybrid is an approach which combines and integrate the methods discussed above and others [23] in attempt to take advantage from the strength of each method. For example cue phrases combined with position and word frequency based methods or position, length weight of sentences combined with similarity of these sentences with the headlines [24].

2.5 Graph Theory Overview

Informally, a graph is a group of points that are connected by a line. The dots are called nodes (or vertices) and the lines are called edges. Graphs are prevalent in Computer Science because they model different kinds of relationships that exist between set of objects. The objects can be programs, people, web pages, sentences and the relationship among them can be dependency, connection, similarity or some other type of relationship. The objects and the relationship among the objects can be modeled by a graph where the objects become the nodes and the relationship among them an edge that connects the two objects in relationship.

Such a graph can be represented in different ways, we can draw it with dots and lines as in ***Figure 2-2***, we can represent it mathematically as a set of nodes and edges, or as an adjacency matrix.

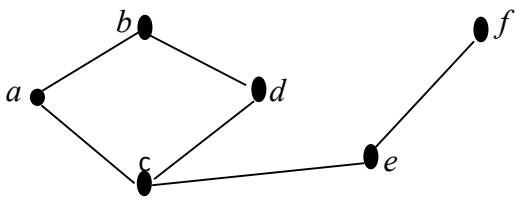


Figure 2-2 : Pictorial Representation of a Graph

Steen [25] defines a graph formally as:

> *A graph G consists of a collection V of* ***vertices*** *and a collection* ***edges*** *E, for which we write G= (V, E). Each edge e ∈ E is said to join two vertices, which are called its* ***end points****. If e joins u, v E V, we write e = <u, v>. Vertext u and v in this case are said to be* ***adjacent****. Edge e is said to be* ***incident*** *with vertices u and v, respectively.*

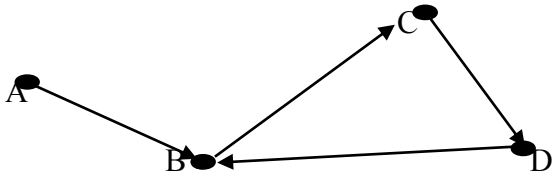
Much often it will be written as V(G) and E(G) to denote the set of vertices and edges with the graph G respectively.

The other common way of representing a graph is as an adjacency matrix, in which the columns and the rows of the matrix represent the nodes of the graph and each element of the matrix represent edges between the nodes.

Formally the adjacency matrix is defined as follows:

Given n-node graph G = (V, E) where V = $\{v_1, v_2 \ldots v_n\}$, the adjacency matrix for G is the **n x n** matrix $A_G = \{a_{ij}\}$ where $a_{ij} = 1$ if $\{v_i, v_j\} \in E$ otherwise $a_{ij} = 0$.

Considering four web pages as nodes and the link among them as edges, the following example in Figure 2-3 shows the adjacent matrix representation of the hypothetical pages and their links.



$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 2-3 : Adjacency Matrix Represenation of a Graph

Graph has been studied in mathematics since the 19th and 20th century [25]. It has introduced different terminologies that allow us to be precise when we describe different characterstics of a given graph. Among the different terminologies that characterize a graph vertex degree, centrality, directed edge and weighted edge are the most pertinent to this research.

- The degree of a vertex is the number of edges that are incident to it.
- Centrality of the node is the measure of the importance of that node in that graph.
- Directed/Undirected Graph – a graph is directed if all the edges are directed from one edge to the other for example if we are representing a one way street from junction A to B, then we need sense of direction, and it is undirected if there is no direction between the nodes relationship.
- Weighted Graph –A weighted graph G is a graph for which each edge e has an associated real-valued number w (e) called its weight [25].

Centrality of a node being the ultimate ranking technique in this work we will discuss four different centrality measurements as presented in [26]. A node's importance, centrality, depends upon what actually the graph models [25]. There are different centrality measurments, among them

- Degree centrality – defines the importance of a node in terms of the number of iteractions, edges incident to it, the higher the degree of the node, the more important the node is.
- Closeness Centrality – defines a node based on how "close" to, and can communicate quickly with, the other nodes in the graph.
- Betweeness Centrality – an important node will lie on a high proportion of paths between the other nodes in the network. This is the less obvious centrality to compute.
- Eigenvector Centrality – an important node is connected to important neighbours in which a nodes importance depends on the importance of the nodes to which it is connected to and the number of nodes it is connected to.

The subsection 2.5 is a prelude intended to give us an overview of the field of graph theory upon which Google's PageRank is founded, Presenting Google's PageRank and its theoretical background becomes relevant to validate the advantage of our new proposed textual graph node ranking algorithm as compared to the use of Google's PageRank for textual graph node ranking.

In the following subsections we will present the computation of Google's PageRank, its adoption for automatic extractive text summarization respectively.

2.6 Google's PageRank

Google's PageRank is one of the most discussed node ranking algorithms due to its success in becoming the founding principle upon which the most popular search engine, Google, is built. The authors of PageRank were motivated by the lack of academic researches on large scale search engines, eventhough the amount of information on the web and the number of inexperienced users were increasing.

These problems were further exaserbated by the cost involved in maintaining high quality human managed indices, indices of keywords versus web pages, and the failure of automated search engines which rely on keyword in providing high quality results, and the vulnerability of key word based search engines for misleaded ranking by advertisers.

PageRank [27] addresses the limitation of search engines by exploiting the additional information present in the hypertext, other than lexical content of the page which was the approach followed by keyword based search engines. They want to answer the question "how to effectively deal with uncontrolled hypertext collections where anyone can publish anything they want"

Bring and Page [27] Answered the stated questions by considering the citation (link) graph of a web that has been largely unused by the keyword based search engines. The main idea in this approach is that – instead of solely depending on the content similiarity between search keywords and a page's content, analyzing the citations among web pages gives us the subjective (As citations are done by people of a given domain, the more a page is cited by more people in that domain the higher the prestige of the page) idea of the importance of the page. This idea is equivalent to the idea that we rank a page based on a collective knowledge of the web community instead of its content, which is hard to fool.

PageRank [27] gives two intuitive justification of their PageRank Algorithm, the first intuitive justification is modeling a user behavior, in which it is assumed that there is "a random surfer" who is given a page at random and keeps clicking on the links in the page, without hitting back and the proabability that a given page will be visited in this model becomes its PageRank.

The other intuitive justification is that a page can have a high PageRank if there are many pages that link to it or if there are some pages that point to it and have a high PageRank.

Google's PageRank can be calculated in what is called Iterative Method or as an Eigenvector of a stochaistic, irreducible and aperiodic adjacency matrix for an Eigenvalue of 1.

The Iterative Method as pointed out in their paper entitled "The Anatomy of a Large-Scale Hypertextual Web Search Engine" [27] is defined as follows

We assume page A has pages T1….Tn which point to it (i.e., are citations). The parameter *d* is a damping factor which can be set between 0 and 1. We usually set *d* to 0.85.Also C (A) is defined as the number of links going out of page A. The PageRank of a page A is given as follows in equation (3):

$$PR(A) = (1-d) + d * \left(\frac{PR(T_1)}{C(T_1)} + \cdots + \frac{PR(T_n)}{C(T_n)}\right) \tag{3}$$

Brin and Page [27] stated that the PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d, various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

To give a theoretical example, if we just consider a world wide web to be just a collection of three pages A, B, C and A is linked to B and C whereas B is linked only to C and C is only linked to B which can be depicted as follows in ***Figure 2-4***.

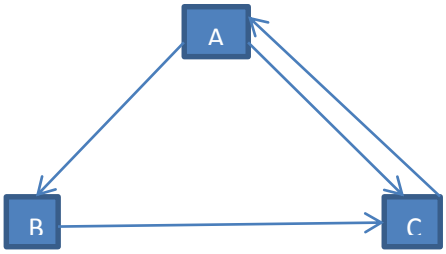


Figure 2-4 : Example of Web Links

The PageRank of each of the above pages using the PageRanking algorithm will be as follows, to keep the calculation simple the damping factor which is generally 0.85 is now assumed to 0.5, so given the above information and the Google's PageRank algorithm their PageRank will be as follows:

$$PR(A) = 0.5 + 0.5(PR(C))$$

$$PR(B) = 0.5 + 0.5\ (PR(A)/2)$$

$$PR(C) = 0.5 + 0.5\ (PR(A)/2 + PR(B))$$

This equation can be solved easily by substitution and the result will be

$$PR(A) = 14/13 = 1.07692308$$

$$PR(B) = 10/13 = 0.76923077$$

$$PR(C) = 15/13 = 1.15384615$$

The higher the PageRank the higher the probability of that page to be accessed by the random surfer who would start clicking at a random page.

The trickier side of the Google's PageRank iterative algorithm is the equation requires that to get the PageRank of a page we need to first get the PageRank of those pages which are pointing to this page i.e. to the page that we want to rank.

Had the pages to be ranked are simple sets of 3 or just few we can use the method of inspection to find the PageRank but since we are considering billions of web links such approach will be prohibitive, and to overcome this hurdle the problem can be approached iteratively by giving equal random values to each page link at iteration of index 0 and doing the iteration until a convergence below a given threshold is achieved. Convergence is achieved when the error rate for the rank of any vertex in the graph falls below a given threshold. The error rate of a vertex V_i is defined as the difference between the "real" score of the vertex S (V_i) and the score computed

at iteration k, $S^k(V_i)$. Since the real score is not known apriori, this error rate is approximated with the difference between the scores computed at two successive iterations: $S^{K+1}(V_i) - S^K(V_i)$ [13].

Considering the above example, the iterative process will result in the following results after 13 iterations.

Iteration	PR(A)	PR(B)	PR(C)
0	1	1	1
1	1	0.75	1.125
2	1.0625	0.765625	1.1484375
3	1.07421875	0.76855469	1.15283203
4	1.07641602	0.76910400	1.15365601
5	1.07682800	0.76920700	1.15381050
6	1.07690525	0.76922631	1.15383947
7	1.07691973	0.76922993	1.15384490
8	1.07692245	0.76923061	1.15384592
9	1.07692296	0.76923074	1.15384611
10	1.07692305	0.76923076	1.15384615
11	1.07692307	0.76923077	1.15384615
12	1.07692308	0.76923077	1.15384615

Figure 2-5 : PageRank Iterative Example

Unlike the simple inspection, the iterative way allows the computation of the rank without being restricted by the size of the nodes even though the computation requires a number of iterations.

The iterative method of PageRank is an alternative explanation of calculating Google's PageRank is using Linear Algebra, by computing the Eigenvector of a stochaistic matrix for the Eigenvalue 1 [28].

The citation among pages can be modeled as a transition matrix A where each element represent the probability of moving from this page to the other which it is citing to. Since the citation is unweighted and directed for a page which cites 5 pages then each page will have an equal probability of 1/4 transition from this page.

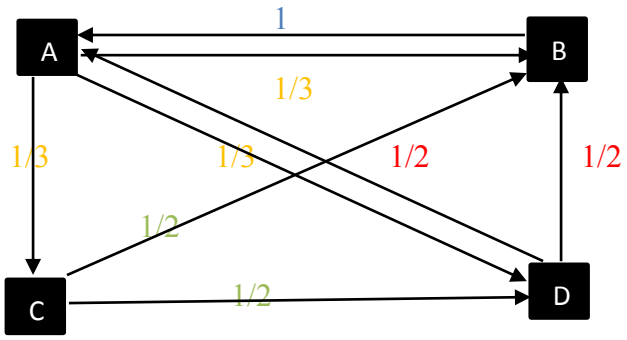


Figure 2-6 : An Example of Web Link References *[28]*

If we consider the above figure as an Internet of four web pages, it can be modeled as adjacency matrix as follows:

$$A = \begin{bmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix}$$

Tansae and Radu [28] States that considering a directed graph and a positive integer *k*. Then the number of directed walks from node *i* to node *j* of length *k* is the entry on row *i* and column *j* of the matrix A^k, where *A* is the adjacency matrix.

The fact that each incoming link increases the importance of a given page, which is the same as multiplying the matrix with a vector which represents the importance of each page, but since the importance of each page is not known in advance, we can start with a vector where all entries are equal to 1/n, where n is the number of nodes.

Considering the above transition matrix A, if we start with a ranking vector where each node has equal importance [¼, ¼, ¼, and ¼], the importance of the pages will be Av, and the importance of each page after one transition will be A^2v, after two transitions A^3v and this continues until the difference between the resulting consequative Eigenvector values are lower than a given threshold.

$$A = \begin{bmatrix} 0 & 0 & 1 & 1/2 \\ 1/3 & 0 & 0 & 0 \\ 1/3 & 1/2 & 0 & 1/2 \\ 1/3 & 1/2 & 0 & 0 \end{bmatrix} \qquad V = \begin{bmatrix} 0.25 \\ 0.25 \\ 0.25 \\ 0.25 \end{bmatrix}$$

$$Av = \begin{bmatrix} 0.37 \\ 0.08 \\ 0.33 \\ 0.20 \end{bmatrix}$$

$$A * Av = \begin{bmatrix} 0.43 \\ 0.12 \\ 0.27 \\ 0.16 \end{bmatrix}$$

This sequence v, Av, $A^2v…A^kv$ iterates and converges resulting in a new Eigenvector, and this new Eigenvector at the convergence is the rank of each node in the adjacency matrix. This is true because If *A* is a positive column-stochastic matrix, then there is a *unique* Eigenvector corresponding to the Eigenvalue $z = 1$ such that it has only positive entries the sum of its entries equals 1 [28].

2.7 The Use of PageRank for Sentence Ranking

PageRank is intended for ranking nodes of a directed unweighted graph which model the hyperlinks among web pages, the rank of each page in PageRank shows the probability of reaching that page through a continuous click without clicking back starting from any randomly selected page.

Another way of looking at this rank is that a page is ranked based on the number of pages that are citing it and the importance of those pages that are citing it [27]. Therefore a graph formed from a lexical similarity between textual units can be analyzed (ranked) using Google's PageRank, two of the works that used this approach are [7, 13]. Once the document is converted into a textual graph, the nodes (the textual units) can be ranked based on the number of other textual units they are linked to and the importance of those textual units that they are linked to.

As explained in the review of related works [13] uses this approach for key word extraction and extractive summary generation. In the extractive summarization generation, sentences are considered to be the textual units (nodes) and the content similarity between them normalized by the length of the sentences is taken as the edge between the nodes. But due to the fact that the edge between textual units is weighted, unlike the edge between web pages which is unweighted, they modified the Google's iterative PageRank to include the weight between nodes in ranking nodes. Given two nodes V_i and V_j, W_{ji} being the weight between the similarity, the modified PageRank [13] is formalized as:

$$WS(V_i) = (1-d) + d \sum_{v_j \in In(v_i)} \frac{w_{ji}}{\sum_{v_k \in Out\,(v_j)} w_{jk}} WS(V_j) \tag{4}$$

And they stated that "While the final vertex scores (and therefore the rankings) differ significantly as compared to their unweighted alternatives, the number of iterations to convergence and the shape of the convergence curve remain almost identical for weighted and unweighted graphs"

On the other hand Erkan and Radev [7] adopt a tf*idf modified cosine similarity with a given thresholding and introduced three different ranking algorithms namely degree centrality, Eigenvector Centrality and Continuous LexRank.

In all the three approaches first sentences with a given level of tf*idf modified cosine similarity value will be removed inorder to consider only those sentences with a strength of simialirty.

The degree based similarity ranks sentences based on their degree, but this approach was discouraged because they believed that it will have a negative effect in measuring centrality when several unwanted sentences vote for each other and raise their centrality.

The Eigenvector centrality which is computed using google's PageRank is chosen over the degree based centrality because a sentence's centrality depends not only on its degree but as well on the importance of those sentences with which it is linked to.

Their third algorithm continuous LexRank, improves their second algorithm by considering the weight of the threshold cut similarity of sentences, and the ranking is also calculated using Google's PageRank by modifiying it to consider the weight of edges.

Even though there is a measure of difference in the way similarities are measured among sentences and the number of alternatives in ranking the nodes, ultimately both TextRank and LexRank use the modified version of Google's PageRank that considers the weight of similarity in ranking nodes.

2.8 Evaluation Method of Text Summarization

Evaluation is an integral part of any undertaking to ensure if the desired objective is achieved or not. Therefore it is necessary to have an evaluation mechanism for automatic text summarization as well which measures the quality of the summary generated using different metrics.

Every evaluation effort works by comparing a given output against a "golden" standard, now the evaluation effort becomes straight forward where reaching upon the golden standard against which other outputs will be compared to is a deterministic issue, but this becomes difficult in text summarization because making the golden standard summary is not a deterministric issue "no one seems to know exactly what a **summary** is" [1]. Not only that but it is also entirely possible that an automatic summary generation system can generate a good summary which is different from the "golden standard".

This is true because sentence selection is not a deterministic problem, i.e. there is no hard and fast rule to dictate which sentence should be included in the summary, and a sentence can be selected or rewritten subjectively and still become acceptable as a good choice. In the case of extractive summary different sentences can be selected by different people or the same person can select different sentences at different time.

Even though finding a golden standard summary against which other summaries can be compared against is a subjective issue, in automatic summarization researches the summary generated by one or more human summarizers will be accepted as the golden summary and the system generated summary will be compared to the single/multiple human summaries using different measures. This kind of summary evaluation is called intrinsic evaluation, which is an evaluation of the system generated summary for its informativeness as opposed to the extrinsic evaluation which evaluates a system generated summary based on how well it can serve when the summary is used as input to other automated information processing.

Steinberger and Jezek [29] Classifies automatic text summarization as Instrinsic and Extrinsic Evaluation. The Instrinsic is further classified into to text quality evaluation and content evaluation.

2.8.1 Intrinsic Evaluation

Intrinsic evaluation evaluates the summary in its own right without considering other external purposes for which the summary can be used. The generated summary is compared with the starndard summary in an effort to assess how much information is preserved in the condensation (Content Evaluation), and based on the linquistic quality of the generated summary (Quality Evaluation).

The amount of information retained in system generated summaries compared to golden standard summaries can be measured based on the exact number of sentences that match between system generated summaries and golden standard summaries (co-selection) or by the relative similarity between system generated summaries and golden standard summaries (text similarity measures).

Three different formulas compute the degree of co-selection in system generated summaries and golden standard summaries, precision, recall, and F-score.

Precision (P) **-** Precision is calculated by dividing the number of sentences that occur both in the automatically generated summary and the standard summary (the intersection of the ideal and generated summary) by the number of sentences in the automatically generated summary.

Ideally the precision measures how many of the automatically generated sentences are identical (precise) with the ideal summary as a percentage of the automated summary. The point here is that, a summary can be precise but not exhaustive, i.e. it is precise but fails to include other sentences in the summary. Because for example, an automatic summary may contain four sentences all which are precise but the ideal summary may contain additional sentences which are not included in the “precise” summary.

Recall (R) **-** Recall is calculated by dividing the number of sentences that occur both in the automatically generated summary and the standard summary (the intersection of the ideal and the generated summary) by the number of sentences in the standard summary.The recall measures the exhaustiveness of the automated summary, because a summary should be precise (match) and also include all the sentences in the ideal summary.

F-score **-** F-score is a composite measure of Precision and Recall which is calculated using the Equation 5.

$$F - Score = \frac{2 \cdot P \cdot R}{P + R} \tag{5}$$

Determining a sentence relevance for inclusion into a summary is, a subjective issue and this subjectivity results in different sentences picked/prioritized by different summarizers. Strictly following co-selection ignores measure of similarities between summaries, because it requires exacting matching between textual units, e.g., sentences, this problem is addressed by using text similarity measurements. Text similarity measurements measure how two textual units or documents are similar, even though there is no exact match of sentences like in co-selection. Text similarity measures measure the similarity between two textual units based on their content overlap even though the two textual units are not totally identical. Cosine Similarity, Unit Overlap, Longest Common Subsequence and N-Gram matches are among other similarity measures used for content similarity evaluation of system generated summaries against golden standard summaries.

Cosine Similarity - when documents or sentences are represented as term vectors the similarity between them can be quantified as the cosine of the angle between the two vectors [30]. Two vectors with the same orientation have a Cosine similarity of 1, two vectors at 90 degree have a similarity of 0, and two vectors diametrically opposed have a similarity of -1. Given two vectors of terms, A and B, the cosine similarity, cos (*θ),* is represented using a dot product and magnitude as shown in Equation 6.

$$similarity = \cos(\theta) = \frac{A \cdot B}{||A||||B||} \tag{6}$$

Unit Overlap - Unit Overlap is the other similarity measure, which computes the percentage of unit overlap as the quotient of the magniture of the lexical intersection divided by lexical union minus the lexical intersection [31] which is formalized as shown in Equation 7:

$$\boldsymbol{Overlap}\,(\boldsymbol{X},\boldsymbol{Y}) = \frac{||\boldsymbol{X}\ \boldsymbol{n}\ \boldsymbol{Y}||}{|\boldsymbol{|X|}|+|\boldsymbol{|Y|}|-||\boldsymbol{X}\ \boldsymbol{n}\ \boldsymbol{Y}||} \tag{7}$$

where X and Y are representations based on sets of words or lemmas. ||X|| is the size of set X.

Longest Common Subsequence

Measures similarity based on their edit distance which is formalized as shown in Equation 8:

$$LCS(X,Y) = \frac{(((length(X) + length\ (Y)) - d(X,Y))}{2} \tag{8}$$

where X and Y are representations based on sequences and where LCS(X,Y) is the length of the longest common subsequence between X and Y, length (X) is the length of string X, and d (X,Y) is the minimum number of deletion and insertion needed to transform X into Y.

N-gram matching (ROGUE)

According to Lin [32] ROUGE is an acronym that stands for Recall-Oriented Understudy for Gisting Evaluation. It is a method to automatically determine the quality of a summary by comparing it to other (ideal) summaries created by humans.

The measure is computed by calculating the content overlap between human summaries (ideal summaries) and the computer generated summaries. Three kinds of ROUGE summaries are known [32] ROUGE – n, which is an n-gram recall between a candidate summary and a set of reference summaries which is calculated using the following formula [32] shown in equation 9:

$$ROUGE - N = \frac{\sum_{S\in\{RS\}}\sum_{gram_n\in S} Count_{match}(gram_n)}{\sum_{S\in\{RS\}}\sum_{gram_n\in S} Count(gram_n)} \tag{9}$$

where RS = Reference Summaries

ROUGE-N computes percentage of the number n-gram matches between the ideal summary and the candidate summary against the total n-grams in the reference summaries. Lin [32] lists the other ROUGE scores, such as ROUGE-L – a longest common subsequence measure and ROUGE-SU$ - a bigram measure that enables at most 4 unigrams inside a bigram components to be skipped.

Intrinsically a summary can also be evaluated for its linguistic quality using human Annotators using four metrics on the scale of A (Very good) upto E (Very Poor) [29] to each summary by considering parameters such as:

Grammaticality – the summary should be grammatical correct therefore grammatical errors like wrong punctuation marks or incorrect words should not be there.

Non-Redundancy – the summary should not contain redundant information.

Referential Clarity – Anaphoric references should be maintained in a summary, there should not be a pronoun in the summary without its correct antecedent.

Structure and Coherence - The summary should show a good structure by maintaining the coherence between sentences.

2.8.2 Extrinsic Evaluation

Extrinsic evaluation is an evaluation of a summary for subsequent process. At times automatic summaries are an input to other information processing activity, instead of being an end by themselves. Hence extrinsic evaluation evaluates the automatic summaries based on their relevance for the subsequent information processing activity for which they become an input, namely these subsequent information processing activities are document categorization, information retrieval and question answering [29].

The other advantage of extrinisic evaluation is giving the degree to which one can be definite in concluding about the usefulness of summaries, which provides a justification for continued research and development for new summarization methods [33].

2.9 The Amharic Language

Amharic is the official working language of the Federal Democratic Republic of Ethiopia. Arguably Amharic is the only sub Saharan language which has its own writing system, making it an ancient leading language in Africa.

From the fact of the Amharic Language being a descendant of the Ethiopic – which will be evident, from a superficial knowledge of both – it claims the same affinity to the Semitic family as its parent; although it has adopted other forms and words from surrounding nation, which bear no relation to that family. Knowledge, therefore, of any of the Semitic Dialects, such as the Hebrew and the Arabic, facilitates, to a great extent, the study of the Amharic [34].

2.9.1 Punctuation Mark in Amharic

Nordquist [35] defines punctuation marks as a set of marks used to regulate texts and clarify their meanings, principally by separating or linking words, phrases, and clauses. Amharic has a total of nine punctuation marks according to [36]. Seven of them are listed below:

1) Word Separator "፡" - unlike English words are supposed to be separated by this mark in Amharic, even though many contemporary writings happen to neglect the use of this mark. In Amharic it is known as – hulet netib.
2) Full Stop (Period) "።" - The Amharic Full stop, or as known in Amharic "Arat Netib "is a serious of four dots which are arranged in square shape.
3) Comma "፣" - The Amharic comma is represented with this symbol.
4) Semicolon "፤" - Also known in Amharic as double cross, or "dirib serez"
5) Preface Colon "፡-" - This is used for introducing speech from a descriptive prefix.
6) Question Mark "፧" - Amharic has quite a different question mark from English, and it is rarely used.
7) Paragraph Separator "፨" - marks the end of a paragraph.

2.9.2 Amharic Grammar

A sentence is an aggregate of words expressing a judgment of the mind [34]. Sentences can be simple, complex, or compound. The constituent parts of every sentence are:

- A subject,
- An attribute,
- A copula or joining verb
- And an object

Among the constituents of a sentence the object is the less necessary than the three other constituents namely subject, attribute and joining verb.

1. **Simple Sentence** is sentence which contains just only subject, attribute and joining verb [34] , for example:

ምድር ሰፊ ናት:: "The Earth is spacious "

ንጉሱ መጡ :: "The King has come"

አንበሳ ፈረስ ገደለ:: "A Lion killed a horse"

2. **Complex Sentences** are sentences which are amplified by qualifying words in connection with either the subject or the attribute [34]; for example:

ልጄ ዛሬ መጣ :: "My son came today"

Is a complex sentence because the subject ልጅ "son" is qualified by the six order sound in Amharic alphabet which changes the word from a single word in to a combination of possessive pronoun + noun i.e. my son.

3. **Compound Sentences** are such as have either the subject, or the attribute, or the object, or all of them, augmented by additional or explanatory parts [34] e.g.

ወታደሩ ፣ ነጋዴዉ ፣ ገበሬዉም የሚጠቅሙ ሰዎች ናቸዉ ።

"The soldier, the merchant, and the farmer are useful men."

Amharic Sentences are constructed with the Subject + Object + Verb combination or with a simple Subject + Verb Combination.

For example the English sentence – Education is a tool for change, more or less can be written in Amharic as "timihirt lelewut mesariya new" and the correspondence between the two will look like as shown in Table 2-1 :

Table 2-1 : Amharic Sentences

English Word	English Part	Amharic transliteration	Amharic Word
Education	Subject	Timihireet	ትምህርት
Is	Verb	Newee	ነዉ
A tool	Object	Mesariya	መሳሪያ
For change	Object	lelewut	ለለዉጥ

To make the Amharic Sentence – we have to join the words in the format Subject + Object + Verb i.e ትምህርት (Subject) ለለዉጥ መሳሪያ (Object) ነዉ (verb).

2.9.3 Amharic Morphology

Amharic words can be morphologically altered by making use of prefixes, suffixes and at times elisions. Both inflectional and derivational morphologies are possible, in the following subsection we will discuss both Inflection and Derivational Morphology for Amharic briefly:

Inflectional Morphology

In Amharic Nouns, verbs, and adjectives can be marked for person, gender, number, case, definiteness, and time [37]. These markings are inflectional morphological alterations of a word which will not alter the category of the part of speech the word initially belongs to. The following examples take the case of Amharic noun to show the different morphological inflections by marking an Amharic noun for person, gender, case and definiteness.

Noun Inflection

1) Noun + Gender Marker Suffix.
2) Noun + Number Marker Suffix.
3) Noun + Case marker Suffix i.e. both for nominative and accusative case.
4) Noun + Definiteness Marker Suffix.

***accusative case** - the case of nouns serving as the direct object of a verb – according to the freedictionary.com

***nominative case** - the category of nouns serving as the grammatical subject of a verb – according to the freedictionary.com

Table 2-2 shows the different Amharic suffixes which can be inflected on some nouns to mark them for gender, number, case and definiteness.

Table 2-2 : Amharic Suffixes

	Suffixes			
Noun	Gender Marker	Number Marker	Case Marker	Definiteness Marker
	ኢት - eat	ኦች - och	ን - ne	ኡ - oo
		ዎች - woch	ዬ - ye'a	ዋ - wa
		ኣን - an	ኤ - ea	ዉ - wu
		እየ - e'eye	ዎ - wo	ኢቱ - itu
		ኣት - at	ሀ - hee	ይቱ - yitu
			ሽ - she	
			ኡ - uu	
			ዋ - wa	
			ኣችን - acheen	

The following example shows the inflection of Amharic nouns with the suffixes in Table 2-2 to mark the nouns for gender, number, case and definiteness. It is worth noting that those inflections will not change the part of speech of the word being inflected i.e. it remains noun in this case.

Gender Marker – the example marks (inflects) the Amharic word for Ethiopia (ኢትዮጲያ) and changes it to a new word which means a female Ethiopian by using the gender marker (“ኢት” – “et”).

ኢትዮጲያዊት - (ኢት) – An Ethiopian, Female.

Number Marker – the example below takes two Amharic nouns student (ተማሪ) and father (አባት) and inflect them with Amharic suffixes for number markers to result in new words students and fathers by using the number markers (“ኦች” - “och”) and (“ዎች” – “woch”). These two number markers change a singular noun into plural and are applied depending on the noun, serving much like the English “s” and “es”.

ተማሪ + ዎች **አባት + ኦች**

ተማሪዎች - (ዎች) – Students. **አባቶች - (ኦች) – Fathers.**

Case Marker – case markers mark a noun to determine the grammatical function performed by the noun, i.e. as subject or object. The following example takes the Amharic words for Mighty (ሓያል) and Country (አገር) and inflects them with different suffixes to mark the noun as subject or object of a sentence.

Casing noun for Subject

ሃያላን: (አን) – Mighty People

Casing noun for Object

አገርን : (ን) – The country.

አገርዎ : (ዎ) – Your country, honorary.

አገርህ : (ህ) – Your country, when you refer to masculine.

አገርሽ : (ሽ) – Your country, when you refer to feminine.

አገራችን : (አችን) – Our country.

Definitiveness Marker: definitive markers are generally used with nouns, to denote entities about which a speaker or writer is confident that the hearer or reader knows about. The following two examples take the Amharic word property (ንበረት) and inflect it with definite markers and change them to refer to entities which are defined earlier.

ንበረት + (ኡ) **ንበረት + ዋ**

ንብረቱ : (ኡ) – His property **ንበረትዋ : (ዋ) – Her Property**

Derivational Morphology

Morphological inflections are called derivational when the inflections changes the category of the part of speech a given words belongs to i.e. by adding a prefix, infix, or suffix or by the process called a elision when a word belonging to one category of parts of speech is transformed to another one.

Noun Derivation

Amharic Nouns can be derived from basic nouns, adjectives, verbs, stems and roots [37]. "ነት" "አት" are used to derive nouns from the basic forms of nouns and adjectives, respectively. The following examples illustrate the use of the suffixes "ነት" "አት" to derive noun studentship (ተማሪነት) from the noun student (ተማሪ) and nearness (ቅርበት) from the adjective near (ቅርብ) respectively.

ተማሪ = ተማሪነት (ነት) – the noun student is changed to studentship.

ቅርብ = ቅርበት (ኧት) – the adjective near is changed to nearness.

Verb Derivation

Unlike other word categories the derivation of verbs from other parts of speech is not common [37]. The Amharic verbal roots have the pattern of three consonants. Some verbs can be derived from such forms by fusing vowels with one or more of the consonants as shown in the example below [37].

ውስድ is changed to **ወሰደ (ዉ-ኧ-ስ-ኧ-ድ-ኧ)**

Adjective Derivation

Adjectives are derived from verbs, nouns, verbal roots, and stems by adding suffixes, with the most common suffixes here including – "ኛ : nga", "ኣማ : ama", "ኣም:am", "ኣዊ : awi" and "ኣ:a". For example the word history (ታሪክ) is changed to an adjective historical (ታሪካዊ) with the suffix "ኣዊ".

ታሪክ (History) - ታሪካዊ (ታሪክ-ኣዊ) (Historical)

Chapter 3
Related Work

Automatic Text Summarization has been researched since 1950 and a large number of researches have been published on the topic, a bibliography of Research in Text Summarization published on 1998 by Association for Advancement of Aritificial Intelligence (AAAI) lists more than 100 researches done in Automatic summarization.

In this Chapter we will review a number of automatic text summarization researches which are conducted both at a global and local level, focusing our selection based on their pertinence to the stated problem the research addresses; exploring the use of graph based sentence ranking for generic and unsupervised automatic extractive text summarization for Amharic document.

3.1 Related Works in English Language

The earliest and pioneering extractive summary generation is done in [3], motivated by the increased computational power of IBM 704 and 705 Computers to calculate statistical features of a document, which are used in the article for measuring the relevance of sentences.

The research measures the significance of a sentence by considering the frequency of the words it contains and the position of a word in the sentence, the researcher puts the following statement in measureing a sentence's relevance, "a sentence's relevance is determined from an analysis of its words. It is here proposed that the frequency of words occurrence in an article furnishes a useful measurement of word significance. It is further proposed that the relative position within a sentence of words having given values of significance furnishes a useful measurement for determining the significance of sentences [3]."

The research has made a significant pioneering contribution in automatic text summarization but it suffers the disadvantage of being dependent on the style of the writer and further more considering all words in calculating for word frequency bears its own negative effect because words like "a", "the" which we call as stop words will have an exaggerated frequency value.

In [4] Edmondson, the immediate sequel to [3], capitalizes on [3] by adding additional three features other than word frequency and their co-occurrence in measuring significance of a sentence.

Namely these additional methods are:

1. Pragmatic Words (Cue Words)
2. Title and Heading Words
3. And Structural Indicators (Sentence Locations)

Even though the additional methods improve the quality of the summary they also restrict the kinds of documents to which the approach can be applied to; because features like title are specific to some kind of documents.And the structural indicators for example "topic sentences tend to occur very early or very late in a document and its paragraphs" [4] are statistically learned feature of certain kinds of documents which are specific to specific kind of documents and learning those features is a supervised task which requires a training corpus.

After the 70's there have not been much work in automatic text summarization, but recently due to the increased electronic data and improvement in computational powers has restarted the interest in automatic text summarization.

Gong and Liu [38] used Latent Semantic Analysis for automatic text summarization being inspired by Latent Semantic Indexing which has been in use in keyword search document retrieval system.

LSA starts by forming the co-occurrence matrix of sentences and words of a document for topic identification, unlike LSI (its predecessor) LSA uses sentences instead of documents as the column of co-occurrence matrix on which the vector Singular Value Decomposition will be used.

The reason the SVD is applied to word-to-sentence co-occurrence matrix is to reduce the dimensionality there by have those words and sentences that have strongest relationship and remove word and sentences that are weakly related. This means the initial matrix will be redrawn with reduced dimensionality while maintaining the most significant co-occurrence, and the bag of words which are identified as having strong relations are used to identify the important sentences for inclusion into the extracted summary.

But reducing the dimensionality is the challenge of LSA because keeping the dimension few results in leaving out important patterns and increasing the dimension results in introducing the noise of the original data [11].

Probabilistic Latent Semantic Analysis (PLSA) is the other approach which is similar to Latent Semantic Analysis (LSA) in modeling topics of a document but unlike LSA, PLSA has a solid statistical foundation as it is based on the maximum likely hood principle and defines a proper generative model of the data [6]. Different researchers has applied PLSA to model the topics (bags of words which are semantically related), for generic automatic extractive summary generation. PLSA uses the latent variable model, in which the latent/hidden variables

(represented by topics/concepts) are associated with the observed variables (represented by documents and words, for the text domain) [12].

Bhandari, Shimbo, Ito and Matsumoto [39] used PLSI in which each document is represented as a term frequency matrix and by using EM-algorithm P(w/z), P(d/z) and P(z) are calculated where w stands for word, d stands for document (in this case sentence) and z the latent variable i.e. the topic. And the equations P (d/z) represent the importance the document d in a given topic (z) represented by P (z). And z with the highest P (z) is picked as the central topic of the document and the sentence with the hight p (d/z) score contained in the selected topic are picked.

This approach of considering only sentences with the highest rank in the highest ranked topic fails to take advantage of the fact that PLSI divides the document into several topics. To make their extract to include sentences that better represent the several topics instead of only the most important topic they introduced a new sentence measure which picks sentences which have good influence ranging over several topics better.

Even though PLSA can be used in generating generic extractive summaries which can range different topics covered in a document, the algorithm requires the determination of the number of topics covered in the document in advance which actually requires an experimental training making it a supervised approach.

The other approach that has grown in popularity in extractive summarization is graph based approach. Central to graph based extractive summarization are the representation of a text as a graph and the use of graph based ranking algorithms to rank the vertices in the graph based on their centrality. A document d, can be represented as a graph G (V, E) in which V represents the set of textual units (sentences, phrases, or words) and E represent the textual relationship between the textual units' interms of similarity.

Mihalcea and Tarau [13] used Google's PageRank for key word extraction and automatic document summarization. Given a directed graph G = (V, E) where V are the set of vertices and E set of edges, and for a given vertex V_i, let **In (V_i)** be the set of vertices that point to it and let **Out (V_i)** be the set of vertices that V points to, using Google's PageRank the rank the vertex V_i will be calculated as follows

$$R(V_i) = (1 - d) + d \left(\sum_{j \in In(V_i)} \frac{1}{|Out(V_j)|} R(V_j) \right) \quad (10)$$

where $R(V_i)$ is the Rank of Vertex V_i and *d* is called a damping factor which usually is set to 0.85 *[27]*.

This formula is bound to be iterative as the rank of a given page is dependent upon the rank of the pages pointing to and there is no way of predetermining the rank of the pointing page.

Google's PageRank considers a directed and unweighted graph which models a reference link among web page, where as a textual graph modeling sentence relationship based on their similarity is undirected and weighted. Even though the direction of the link doesn't have any effect on the calculation of the rank, the weight of the relationship has therefore [13] modified Equation 3 to consider the weight of the edge connecting the two nodes, and they have claimed that the consideration of the weight will result in different ranking but that is appropriate while considering the weight factor, the only negative effect is, the weight increases the number of iteration. The weight modified PageRank is formalized as follows:

$$WR\,(V_i) = (\,1 - d) + d\left(\sum_{j\,\in\,In(V_i)} \frac{w_{ji}}{\sum_{v_k\,\in Out\,(v_j)} W_{jk}}\; WR(V_j)\right) \tag{11}$$

where w_{ji} represent the weight (similarity) between vertex v_i and its adjacent sentences.

Mihalcea and Tarau [13] defined the connection between two sentences to be their similiarity where similarity is measured as the degree of content overlap between the two sentences, moreover to avoid favoring longer sentences they use a normalizing factor and divide the content overlap of the two setnences by the length of each sentence. Which is formally given as follows, given two sentences S_i and S_j, with a setnences represented by the set of N_i words that appear in the sentence : $S_i = W^i_1, W^i_{2,\ldots,} W^i_n$. The similarity of Si and S_j is given by equation (12):

$$Similarity\,(S_i, S_j) = \frac{|\{\, w_k \in S_i \;\&\; w_k \in S_j\}|}{\log(|S_i|) + \log(|S_j|)} \tag{12}$$

The textual graph then is constructed from the similarity matrix that is built from calculating the the similarity between each sentence in the document using the above formula, which will result in a highly connected graph.

The modified Google's PageRank algorithm, modified to consider the weight of similarity will be applied on the subsequent graph to rank each node (sentence), after which sentences are sorted in descending order based on their rank and the top ranked sentences are selected for inclusion in to summary.

The other work which follows a similar approach with [13] is the work in [7] which proposes a new approach called LexRank for ranking sentences based on the Eigenvector centrality in a graph representation of sentences.

Erkan and Radev [7] Represent cluster of documents as a cosine similarity matrix where each entry in the matrix is the tf*idf modified cosine similarity between the corresponding sentences. After this representation they define a threshold to avoid sentences similarity below the

threshold, but this approach of setting a threshold is purely experimental which makes the process supervised. After the thresholding they propose the following ranking algorithms:

Degree centrality – in which sentences are ranked based on their number of connections.

Eigenvector Centrality or LexRank – in which they followed google's random surfer model which ranks each node in the graph based on the probability that a given surfer will reach at that node without being stuck in any part/cluster of nodes by randomly starting from any given node in the graph.

The rank of their probability is represented by the Eigenvector that will be iteratively computed from the similarity matrix which is modified to be stochaistic, aperiodic, and irreducible for an Eigenvalue of 1.

To make the matrix stochaistic i.e. a matrix where the sum of rows is equal 1, a given sentences will distribute its degree among its neighbours equally which forms the stochaistic matrix B, represented by the equation 13:

$$B\ (i,j) = \frac{A\ (i,j)}{\sum_k A(i,k)} \tag{13}$$

where each element is divided by the row sum (degree) of that row (node or sentence).

To make the stochaistic matrix irreducible and aperiodic, i.e. a markov chain is irreducible if any state is reacheable from any other state and it is aperiodic if the return to a state is reached only at multiple of 1 steps [27] introduced the dumping factor called d which gives every state a default probability of transition d.

This modification of the stochaistic matrix to aperiodic and irreducible is required because of Perron-Frobenius theorem, such a matrix is guaranted to converge to a unique stationary distribution i.e. vector starting from a given uniform value where $X^n(i, j)$ gives the probability of reaching from state i to j in n transitions.

Which can be calculated as the Eigen vector which converges after n transitions starting from a uniform values or by what is called the power method which are given by the following equations 14 and 15.

$$p = [dU + (1 - d)\, B]^T p \tag{14}$$

where U is a unit matrix introduced to make the original stochaistic matrix irreducible and aperiodic, in which each entry in the matrix is 1/n, n is the total number of sentences in a document.

Alternatively each page can be ranked iteratively by the following equation 15:

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in adj[u]} \frac{P(v)}{\deg(v)} \tag{15}$$

Continuous LexRank – similar to LexRank but considers the weight of similarity between the sentences and the corresponding power method formula is modified as shown in equation 16:

$$p(u) = \frac{d}{N} + (1 - d) \sum_{v \in adj[u]} \frac{idf\ modified\ cosine(u, v)}{\sum_{z \in adj[v]} idf\ modified\ cosine(z, v)}\, p(v) \tag{16}$$

3.2 Related Works in Amharic Language

Melese Tamiru [9] is an Amharic summarization research which applies the Latent Semantic Analysis for topic modeling as primary technique to generate automatic text summary and based on the LSA the work proposes two methods for summarization. The two methods proposed are called TopicLSA and LSAGraph.

TopicLSA employs LSA to identify the main topics (bags of words) of a document, then the identified topics cosine similarity with each sentence in the document coupled with document genre information, namely, the position of a sentence in the document and the similarity of a sentence to the title are used in aggregate to rank each sentence in the document.

By LSAGraph the work argues that using semantic representation of sentences avoids the problem of polysemy and synonym which it points out as the problem of previous graph based ranking algorithms used by previous researches such as [13, 7], and it also argues that by doing so text summarization is promoted from key-word level analysis to semantic level analysis. Again the paper states that "we constructed graphs where the nodes in the graph are represented by the semantic representation of the sentences and the cosine similarity between them establishes the edge between nodes."

The LSAGraph has the following two limitations:

i. Graph based ranking is applied after after SVD and the corresponding dimension reduction, therefore the inherent problem of LSA in dimension reduction remains.
ii. The graph based ranking used uses the same iterative ranking algorithms PageRank and HITS for ranking the sentences which this thesis proposes to minimize.

Eyob Delele [10] is another Amharic summarization which proposes the use of Probabilistic Latent Semantic Analysis (PLSA) which is another topic modeling statistical tool.

The thesis proposes six sentence ranking algorithms once the topics are identified using probabilistic latent semantic analysis on word-to-sentence matrix.

They use the six suggested sentence ranking algorithms to rank relevance of sentences to the identified topics.

The sentence ranking in this work estimates the presence of keywords in each sentence and since they experiment on news articles "the sentence ranking algorithm is designed to ALWAYS select the first sentence of the document in order to take advantage of the fact that important information exists near the beginning of news articles most of the time" [10].

The need for the determination of the number of topics in advance by PLSA and the mandatory inclusion of the first sentence in extracting summary sentence is the citable limitation of [6] which makes it a supervised, generic-specific summary which is not portable across all domains.

Another work on Amharic summarization which is different from the other reviewed related works is [8] which considers multi-document summarization, and the main challenge of multi document summarization is redundancy of information in the group of documents to be summarized.

Apart from increased redundancy and large corpus to process the central idea in extracting the summary remains the same, i.e., identify important topics (by avoiding repetition) and extract those sentences which are similar with the topic identified.

The topic from the multi document is identified by a two-step process

1. Sentence to sentence matrix is built and the cosine similarity between the sentences is calculated, which will result in sentence to sentence graph. And those nodes/sentences which are loosely coupled with the rest of the document will be removed resulting in a set of sentences which are highly related.
2. And the remaining sentence matrix which is generated in step one is given to the topic modeler, for selecting the important sentences.

Once the topics are modeled then the research uses four sentence scoring methods for selecting summary sentences.

The proposed sentence ranking methods are:

- ORST (Overall relevance of sentence across the topics)
- GSSST (Graphical Sequential Selection of sentences from all Topic),
- NGSST(Non-Graphical Sequential Selection of sentences from all Topic)
- Topic Fold

This work, since it primarly depend on PLSA, the inherent limitation of PLSA will also be true for this work.

Chapter 4
The Proposed Approach

In this Chapter we discussed the new proposed algorithm. Having a good understanding of general graph theory and Google's PageRank facilitate the discussion of the proposed algorithm. The proposed approach has three distinct measurements Content Overlap (CO), Independent Rank (IR) and Sentence Rank (SR) which allows an improvement over PageRank when used for extractive text summarization.

4.1 The Proposed new Ranking Algorithm

PageRank is a centrality measurement algorithm in the vernacular of graph theory, among the different kinds of graph centralities PageRank measures the Eigenvector centrality, which is presented by one of the different intuitive explanation for PageRank (PageRank measures the importance of a node based on its degree, number of citations, and the importance of those pages which are inDegee with the page).

The idea of Eigenvector centrality is further explained in [13] by the example of "voting" or "recommendation" in which when a node links to another node, it is basically casting a vote for the other node therefore the higher the number of votes are casted for a vertex the higher the importance of that node is. Moreover, the importance of the node casting the vote is taken into consideration by the ranking model. In general the ranking score of a given node is the composite of the number of votes cast for it, and the score of the nodes casting these votes.

The adoption of Google's PageRank in analyzing/ranking textual graphs has the following two limitations:

i. In their article Brin and Page [27] stated that "…PageRank extends this idea by not counting all links from all pages equally, and normalizing by the number of links on a page", from this statement it is evident that an incoming link's importance is reduced by the normalization, but this normalization is not necessary as a textual graph is undirected and therefore no need of normalizing it for its outbound links.
ii. Whichever technique, the iterative process or Eigenvector computation, is used to compute PageRank, it is an iterative process which is iterated until a certain convergence. On a preprocessed matrix PageRank has

polynomial complexity which is shown with the big of notation as **O (n^c)** where c is a variable dependent on three factors namely, the size of n, convergence value chosen and the weight factor considered.

Out proposed algorithm introduces two new measures called Independent Rank and Sentence Rank which can be calculated in a maximum of quadratic order complexity, resulting in an observable improvement over the polynomial complexity of Google's PageRank, and the proposed algorithm delivers a better informative summary which is evidenced by an improved ROUGE-1 result compared to the TextRank and LexRank which are based on Google's PageRank.

4.1.1 Independent Rank (IR)

Eigenvector centrality is a measure of centrality where a nodes importance is determined by:

1. The number of nodes connected to it and
2. The importance of those nodes connected to it

Therefore we introduced a new measurement called Independent Rank (IR), which ranks a single sentence's importance in a given document based on the product of its degree and the weight of its edges. The intuitive explanation to the Independent Rank is that, a sentence is important if it shares similar content with many sentences and the extent of similarity is higher. This measure doesn't consider the rank of the sentences to which it shares a content, but it determines the importance of the sentence in the document independently by itself based on its degree and weight of its edges.

We, as well, proposed that for the effectiveness of the IR a new sentence similarity measure that measures the similarity between two sentences in terms of the content overlap between the two sentences is necessary. As the IR ranks a sentence based on the product of its degree and the weight of its edges, factors like the length of the sentence should not be considered because a sentence is complete in itself, representing a complete concept and the preprocessing has reduced the sentence into a bag of content bearing stemmed words, and this bag of content bearing stemmed words (which embodies a given concept) should be evaluated based on their lexical cohesion with other sentences in the document not based on the length of the sentence from which they are extracted nor the length of the sentences to which they are sharing contents.

Our sentence similarity measure, known as, Content Overlap (CO) is formalized as follows, given S_i and S_j in a document and W_k being the total number of words in the document after stop words are removed and the remaining words are stemmed, the Content Overlap (CO) between two sentence is given by equation 17:

$$CO\left(S_i, S_j\right) = \left\{w_k | \, w_k \in S_i \,\&\&\, w_k \in S_j |\right\} \tag{17}$$

And the Independent Rank is formalized as follows, given Adj (S_i) the set of sentences which are adjacent to S_i by sharing a content with S_i, and $S_{1...N}$ being all the sentences in the document, the Independent Rank of a sentence is given by equation 18 :

$$IR(S_i) = \sum_{S_k \in Adj\,(S_i)\,\&\&\,k=1}^{n} CO\,(S_i, S_k) \tag{18}$$

Calculating the Independent Rank of a sentence in such a manner, gives us the advantage of avoiding the iterations in computing the rank of sentences (i.e. each sentence's/node's rank is calculated **c** times until convergence as presented in the big notation **O (n^c)** until convergence).

The iterative method of Google's PageRank computation is iterative because, the rank of a page is dependent upon the rank of those pages pointing to it and there is no way of predetermining the rank of those pages pointing to this page, therefore the problem is solved by iteratively calculating the rank of each node/page starting with a given equal value, say 1, until convergence.

But because a textual graph is a weighted graph, we proposed that each sentence can be ranked first by aggregating the product of its degree with its similarity. And the Independent Rank can be computed as part of the similarity matrix computation by aggregating each row.

Programatically, the computations are carried out in the following manner. As we have discussed in the literature review part a graph can be represented as a matrix, and a textual graph can be represented as a similarity matrix where both the rows and columns represent each sentence in the document and the entries at the rows and column intersection, except where rows and columns value are the same, will be the similarity of the sentences computed using our newly introduced lexical sentence's content overlap CO.

Given a document D with N number of sentences the matrix representation of the document based on their *CO* will be as shown below:

$$D = \begin{bmatrix} CO_{ij} & \dots & C_{iN} \\ \vdots & \ddots & \vdots \\ CO_{Nj} & \dots & C_{NN} \end{bmatrix}$$

Once the document is represented as a similarity matrix using CO, the IR of each sentence will be the sum of its corresponding row in the matrix representation as shown by the formula (18). A sample document taken from DUC 2002 and the corresponding IR of each the sentences is shown below. The numbers preceeding each sentence are their sequence in the document followed by their IR computed accordingly after the sentences are preprocessed.

1(16):BC-Hurricane Gilbert 09-11 0339.

2(16):BC-Hurricane Gilbert,0348.

3(23):Hurricane Gilbert Heads Toward Dominican Coast.

4(0):By RUDDY GONZALEZ.

5(1):Associated Press Writer.

6(7):SANTO DOMINGO, Dominican Republic (AP).

7(38):Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.

8(12):The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

9(7):"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.

10(11):Cabral said residents of the province of Barahona should closely follow Gilbert's movement.

11(8):An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo.

12(24):Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.

13(31):The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

14(27):The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.

15(12):The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

16(27):Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast.

17(1):There were no reports of casualties.

18(17):San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.

19(22):On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast.

20(8):Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.

21(13):Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.

22(13):The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

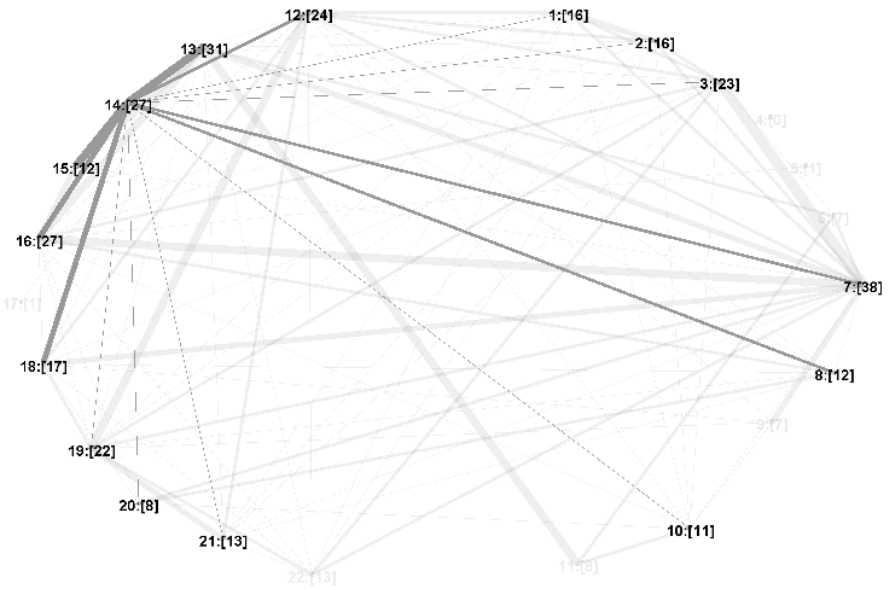


Figure 4-1 : Sample Document Graph with IR

We can see from the weighted graph representation of the Indepenent Ranks of the above document, sentence 14 and sentence 16 are equal with a value of (27) as shown in the figure Figure 4-1. Sentences 14 shares contents with sentences (1,2,3,7,8,10,12,13,15,16,18,19,20,21) a total of 14 sentences and an IR of (27) and sentence 16 shares contents with sentences (1,2,3,5,7,8,10,12,13,14,15,18,19,20,22) with 15 sentences and an IR of (27). This equality in IR was possible despite the different set and different number of sentences adjacent to them (their degree), because Independent Rank considers the weight of the edges in addition to the number of edges.

4.1.2 Sentence Rank (SR)

The Independent Rank (IR) ranks sentences without considering the importance of sentences to which a sentence shares a content, Sentence Rank (SR) considers the importance of the sentences to which a sentence shares a content which is represented by the sentences Independent Rank (IR).

The Sentence Rank (SR) is formalized as follows, given Adj (S_i), the set of sentences which are adjacent to S_i by sharing a content with S_i, and $S_{1...N}$ being all the sentences in the document, the Sentence Rank of a sentence is given by equation 20:

$$SR(S_i) = \sum_{S_k \in Adj\,(S_i)\ \&\&\ k=1}^{n} IR(S_k) * CO\ (S_i, S_k) \tag{20}$$

The advantage of Sentence Rank (SR) is that it can be computed with a maximum of quadratic complexity of O (n^2) where n is the total number of sentences in a document, as the Independent Rank (IR) can easily be computed as part of the similarity matrix preparation by aggregating the CO values of a row (sentence). This computation of Sentence Rank reduces the complexity of ranking individual nodes from a polynomial order of the Google's PageRank to a maximum of quadratic complexity.

Ardo [40] presents the pseudocode for Google's PageRank which we have shown below:

```
proc PageRank (G Web Graph, q damping factor ===0.15)
  N <-- |G|
  for each p ∈ G do
      Pagerank = 1/N
      Aux_p = 0
  od
  while (PageRank not converging) do
      for each p ∈ G do
        r+(p) <-- pages pointed by p
        for each p' E r+(p) do
            Aux_p' = Aux_p' + Pagerank_p / |r+(p)|
        od
      od
      for each p ∈ G do
        Pagerank_p = q/N + (1 - q) Aux_p
        Aux_p = 0
      od
      Normalize PageRank: ∑PageRank_p = 1
  od
end
```

We can see from the pseudocode for PageRank that it receives the web graph and the damping factor as input and start by giving an equal rank of 1/N for each page, then it iterates a number of times to reach to the rank of each page until convergence i.e. shown by the condition **while** (PageRank not converging) **do** in the pseudocode. It is this convergence iteration, that with our proposed approach, will be avoided and each sentence will be ranked just once.

Our attempt to reduce the number of iterations doesn't consider the complexity inside the while loop, since our objective is reducing the number of convergence iterations. Even in that case, the total complexity of our Sentence Rank (SR) and Independent Rank (IR) is much lower than that of PageRank as shown by the pseudocode presented below for our Sentence Rank (SR) and Independent Rank (IR).

```
proc SentenceRank (G Text Graph)
      for each s E G do
        r+(p) <-- corresponding weight of sentences adjacent to s
        for each p' E r+(p) do
            IRp' = |r+(p)|
            SRp' = SRp' + (COp' * IRp')
        od
      od
end
```

In addition to the iteration improvement, the use of the new algorithm extract better representative sentences compared to LexRank and TextRank which uses PageRank for ranking the sentence nodes, this is evidenced by an improved ROUGE-1 result on 2002 DUC dataset and 30 Sample Amharic documents.

Considering the same sample document the document is modified to show the sequence of sentences with their corresponding Sentence Rank value. The number before the bracket is the sequence of the sentence and the number in the bracke is its Sentence Rank.

1(256):BC-Hurricane Gilbert 09-11 0339

2(256):BC-Hurricane Gilbert,0348

3(506):Hurricane Gilbert Heads Toward Dominican Coast

4(0):By RUDDY GONZALEZ

5(1):Associated Press Writer

6(49):SANTO DOMINGO, Dominican Republic (AP)

7(1406):Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas

8(144):The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph

9(42):"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday

10(121):Cabral said residents of the province of Barahona should closely follow Gilbert's movement

11(64):An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo

12(576):Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night

13(837):The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo

14(729):The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm

15(144):The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday

16(702):Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast

17(1):There were no reports of casualties

18(289):San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night

19(440):On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast

20(64):Residents returned home, happy to find little damage from 80 mph winds and sheets of rain

21(169):Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane

22(169):The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month

Once the Sentence Rank is calculated the sentence will be sorted in descending order based on their Sentence Rank, and sentences with highest rank will be considered for summary based on the percentage of summary requested by the user.

Sentence Rank differs from Independent Rank because it considers the importance of those sentences to which it is connected to, i.e., a sentence which shares a content with more important sentences will have a higher Sentence Rank. We show the difference between Sentence Rank and Independent Rank using the graphical representation of the Sentence Rank and Independent Rank of the above document in Figure 4.1 and Figure 4.2 respectively.

Sentences 14 and 16 even though they have equal Independent Rank their Sentence Rank is different because Sentence Rank considers the importance of (in our case the Independent Rank) of those sentences to which it is connected and the weight of (the number of contents they share).

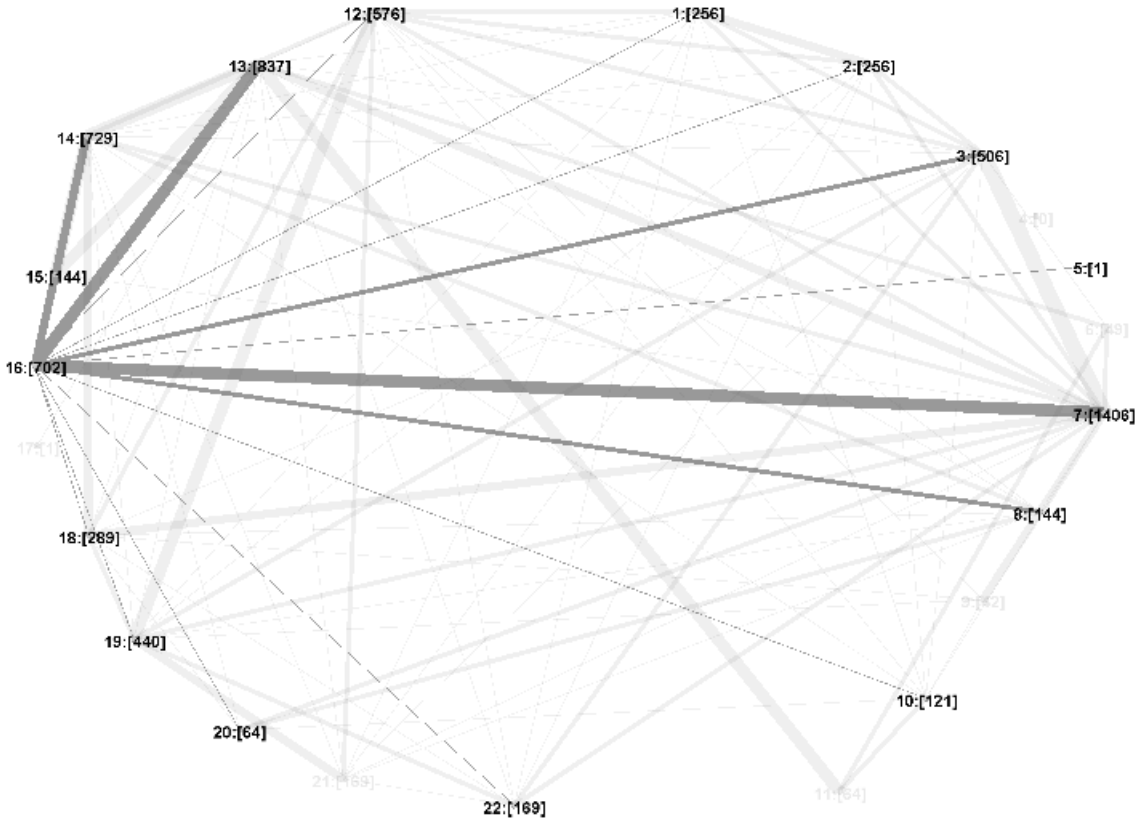


Figure 4-2 : Sample Document Graph with SR

We can see from the Sentence Rank weighted graph representation of the document that sentence 14 has a Sentence Rank of [729] and sentence 16 a Sentence Rank of [702]. The point here is that even though both sentences have equal Idependent Rank and sentence 14 has a lesser degree of 14 its final Sentence Rank exceeds that of 16 by a value of 27, the difference comes from the weight of similarity that sentence 14 have with sentences such as 16 and 18 and as well with other important sentences.

Another additional point that is worth considering from this graphical representation of the document is the characterstics of sentence 3 and 17, as that validates our new similarity measurement CO to effectively rank sentences using IR and SR of a sentence as a sole measure of its importance without factors like the length of sentences. As we can see sentence 3 being a topic sentence its importance in the document is evident, and that fact is shown by our approach, as sentence 3 is ranked above much longer sentences such as 11,18,19,22 in the document. This is possible because each sentence is evaluated based on its IR and SR i.e. its lexical cohesion with other sentences and the importance of those sentences to which it's overlapping.

Similarily we see that sentence 17 which is short yet have a very weak cohesion with sentences in the document is ranked lowest, only better than sentence 4 which actually doesn't share any content with any other sentence in the document.

We believe that our proposal, that a sentence represent a complete concept by itself, regardless of its length and therefore each sentence should only be evaluated by its cohesiveness (IR and SR) in a document is proven by the example of these 2 sentences rank using our proposed approach.

4.2 Prototype Design and Implementation of Sentence Rank (SR)

The Prototype is designed with three main components, the preprocessing component, the summarization component, and the ROUGE-N component.

The preprocess component is a language dependent component. Essentially the preprocessing involves removing stop words and stemming content words, and these tasks are language dependent because we need to have a list of stop words to remove stop words and we need a language specific stemming procedures.

The summarization component takes the preprocessed document as a set of sentences which are reduced into a set of content words which are stemmed and ranks each sentence based on their degree, the weight of similarity and the importance of those sentences which are incident upon it using our newly proposed algorithms, namely, Independent Rank (IR) and Sentence Rank (SR). This component is language independent because sentences are ranked solely based on their lexical relationship.

The ROUGE-N component evaluates the generated extractive summary against two ideal summaries for that specific document, ROUGE-N being a co-occurrence statistics, it measures the amount of content term overlaps between the system generated summary over the set of ideal summaries supplied.

Operationally the preprocessing, as shown in Figure 4-3 (for stop word removal) and Figure 4-4 (for content word stemming), begins by reading the document, once the document is read it will be tokenized into sentences, and then each sentences is further tokenized into a set of words in the sentence. Then each sentences, and each word in each sentence will be preprocessed iteratively i.e. if it is stop word it will be removed and if it is not stop word it will be stemmed, finally words of the sentence that are non-stop word and are stemmed will be used to reconstruct to form the preprocessed sentence and the preprocessed sentences stringed together to form the preprocessed document.

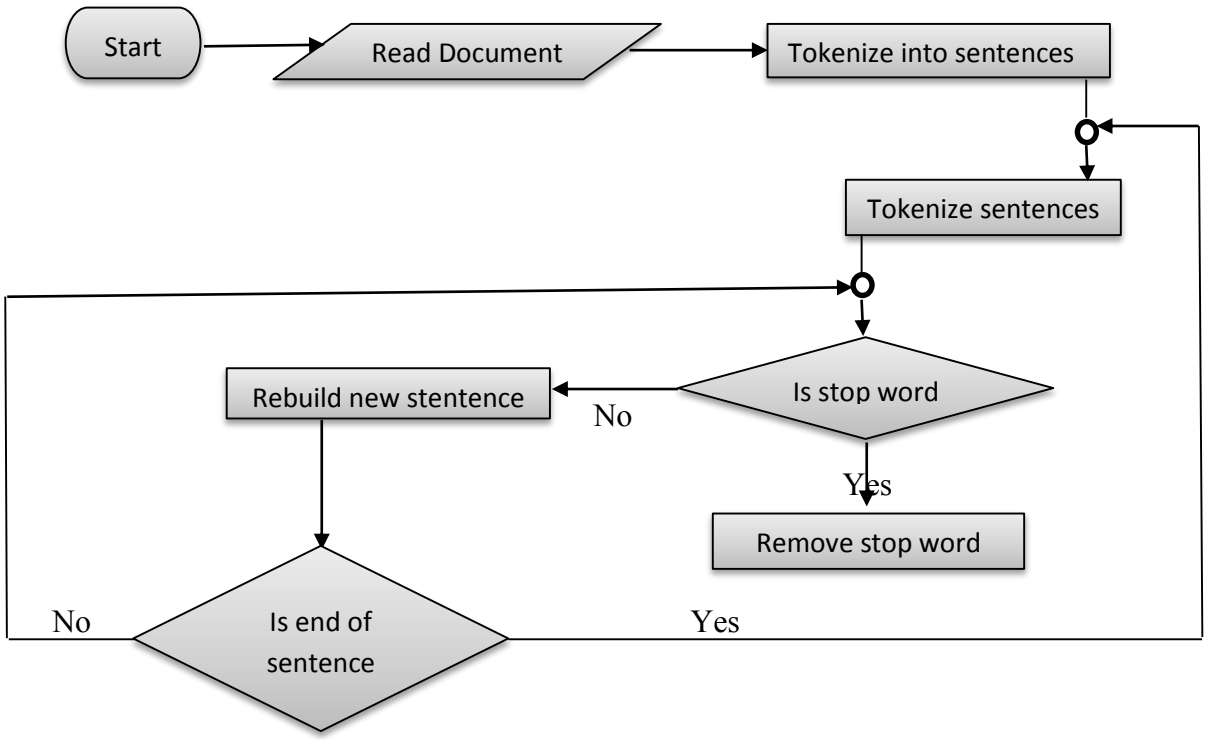


Figure 4-3 : Stop Word Removal

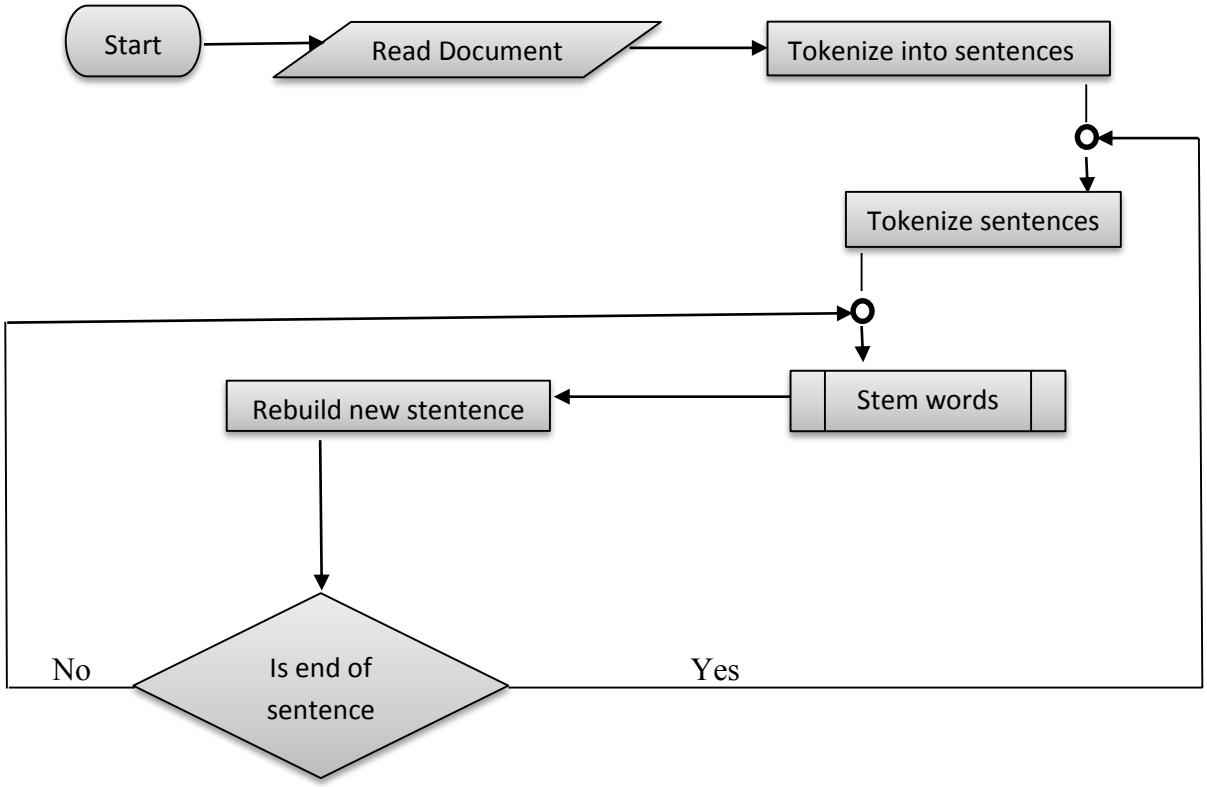


Figure 4-4 : Stemming

Once the document is preprocessed, the summarization module as show in figure 4-5, creates a sentence-to-sentence similarity matrix. The entries of the matrix being the Content Overlap between preprocessed sentences which is done by the compute content overlap method, and followed by aggregation of each CO for a sentence into its IR, then SR for each sentence is computed by aggregating the product of the CO and IR of that sentence.

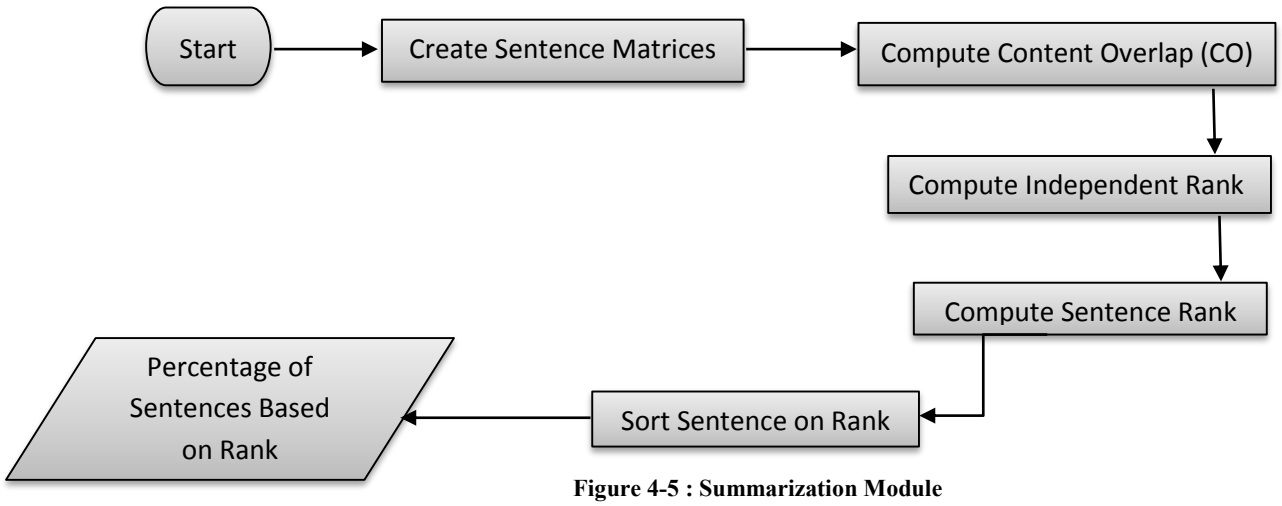


Figure 4-5 : Summarization Module

The evaluation module computes the ROUGE-1 value of the system generated summary against the set of ideal summaries supplied. Our prototype considers two ideal summaries supplied by two human summarizers. Therefore, the evaluation module takes three summaries which are the summary generated by the system and two ideal summaries supplies as reference summaries then the 1-gram co-occurrence statistics between the system generated summary and the reference summary sets will be computed by the module as shown in figure 4-6.

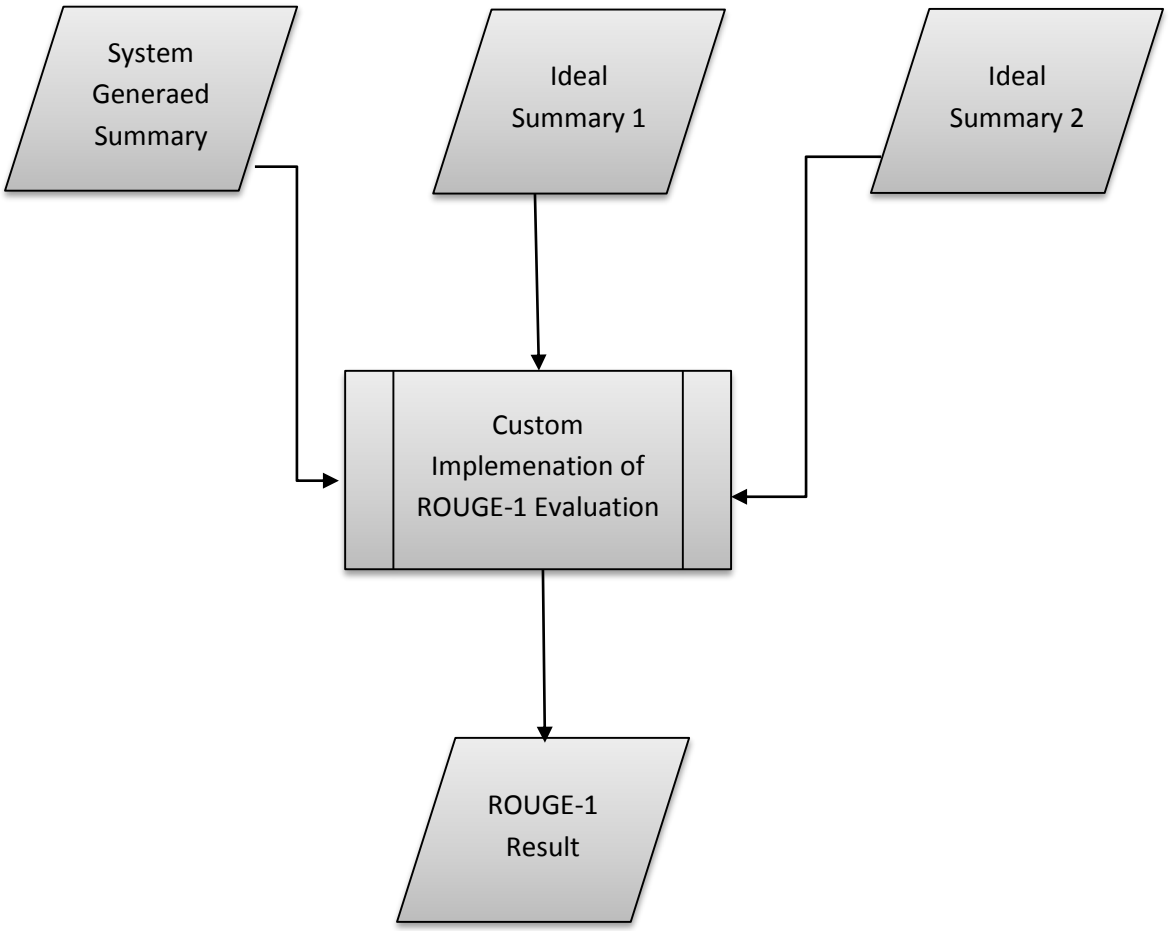


Figure 4-6 : Evaluation Module

4.3 Prototype Screen Shot and Guideline

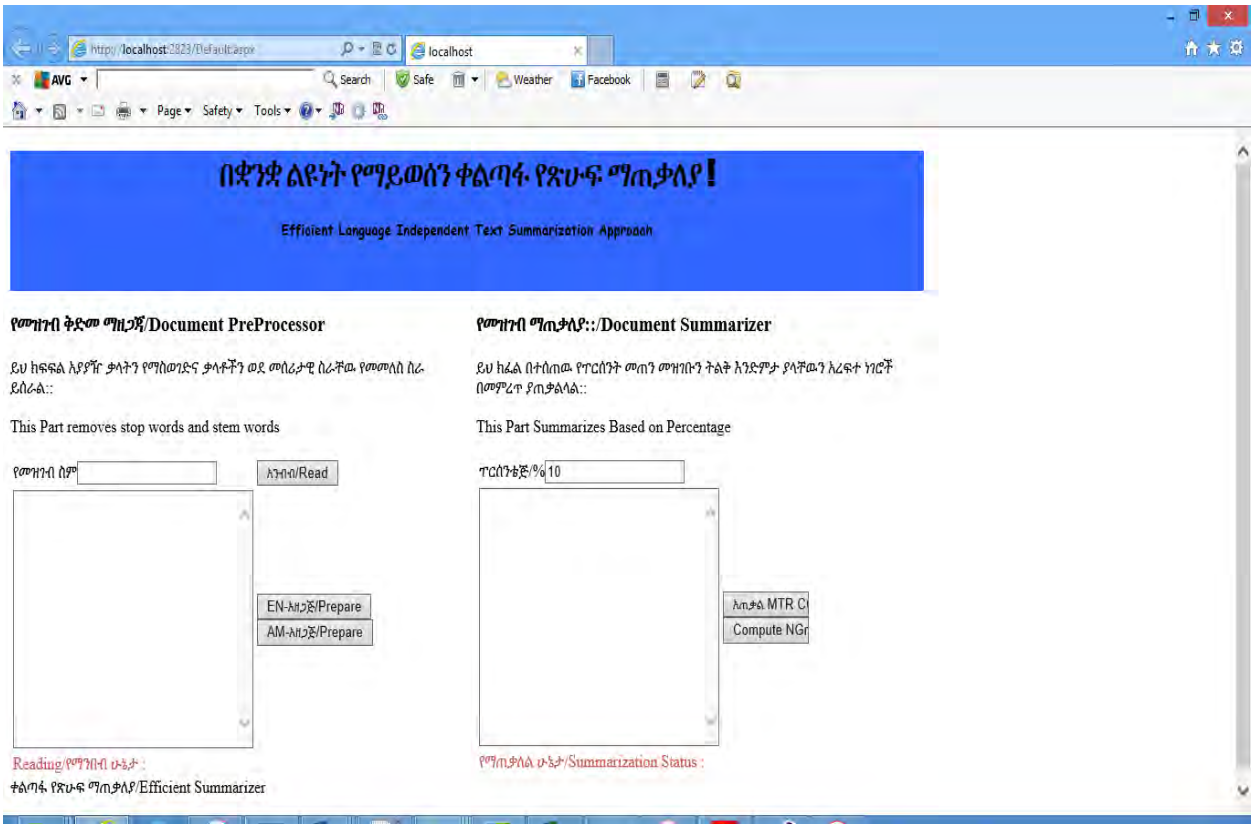


Figure 4-7 : Prototype Screen Shot

The prototype, the screenshot displayed on Figure 4-7, reads the document to be summarized, preprocess it in a language dependent manner and finally evaluate the system generated summary against the set of ideal summaries. The documents to be summarized should be place in the InFolder with .txt extension.

Once the documents are read, after providing the full file name, the document need to be preprocessed by selecting the appropriate preprocessing language by the document's language. Since preprocessing is language dependent the current prototype considers only two languages Amharic and English.

The summary generation requires the percentage of the summary. Therefore, to generate the summary we have supply the percentage of extraction which will be later normalized to the minimum of hundred word (count of words in the selected sentences) set of sentences.

Finally if we supply the set of ideal summaries in the input folder we can compute the ROUGE-1 result of the system generated summary against those set of ideal summaries.

Chapter 5
Experiment

In this Chapter we present the experiment carried out in evaluating the performance of the new proposed algorithm and its comparative result compared with TextRank, the graph based extractive summarization algorithm.

5.1 Experimental Setting

Tokenizing a document to be summarized into a set of distinct sentences is a compulsory task for identifying individual sentences of a document, and this task require a means of identifying sentence ending characters.

Considering "." as a sentence ending characters, for the case of English data, will create a confusion for computerized sentence tokenization as full stop or period serves multi-purpose in English language, for example "." can be used to show decimal places as in the sentence "The temperature is 22.3 degree centigrade" or as shortening of some words such as Doctor to Dr. among other purposes it can serve. Therefore, for successful tokenization either we need an XML representation of the document to be summarized and a corresponding XML parser, as the practice in DUC researches or we need a manual means of marking each individual sentence in the document.

For our test purpose, we manually modified each English document with a special Unicode Character to serve as the end of a sentence. In fact, we could also have stipulated that each sentence be on a new line but such approaches further complicate the manual reviewing and evaluating of a document as there are some sentences which are very long in documents.

Amharic has a very distinct sentence end marker "።" which is known as "Arat netib" but due to the typing complexions, as there are no full featured Amharic editors, many contemporary literatures join two colons ":" as "::" to form the Amharic "።" and others join the Amharic colon "፡" to form "።". Such modifications further complicate the task of sentence tokenization as each of the above Amharic period morphing has their own separate Unicode representation and the Amharic ":" or "፡" also serve as word separator in Amharic.

We alleviated this problems by stipulating that:

1. All words should be separated by space, not by the English or Amharic colons i.e. ":" or "**፡**".
2. Sentences can end in either Amharic period "**።**" or its customizations "::" or "**፡፡**" which formed by joining the English or Amharic colons, the only problem associated with this is that empty sentences will be introduced, which will be further cleaned by an empty sentence removing routine.

5.1.1 Experimental Data Source

Totally there are 527 documents in the DUC 2002 dataset, of which 51% of them (264) are used for evaluating the proposed algorithm.

The Amharic language summarization is tested on 30 Amharic articles collected from four major news outlets namely Ethiopian News Agency, Walta Information Center, Ethiopian Reporter Newspaper and Addis Admas News Paper.

5.1.2 Tools Used For Implementation

- The Algorithm is implemented using ASP.Net and C# which is built on the .Net Framework 4.5.
- Given the fact that the web application is built using Visual Studio 2013, it is deployed and accessed from Microsoft IIS Express Web Server.
- The File For summarization read from a local hard disk in the location as specified by the configuration guideline.

5.1.3 Configuration

The configuration or system setting is required to give the web application the default directory from which it will pick the input file to process, and where necessary the default location where the final output should be stored.

- InFolder – The folder from which the Input files is read.
- OutFolder – The folder to which the final output i.e. the summary is written.

5.1.4 Experimental Data Summary Generation

The DUC 2002 summaries are manually created summaries by a human summarizer. By manually created summaries it is implied that the summaries are abstracts instead of extracts.

The summaries of each document set are provided with a folder named in the format dnnnx:

Where d – means document

nnn – a document number from 061 – 120

x – a placeholder for the first letter of the name of the assessor who generated the summaries.

In each folder there will 7 kinds of different summaries generated by the assessor, which are

1. 200e and 400e – 200 and 400 word extracts for current document set, this is a multidocument kind of summary.
2. 10, 50, 100, 200: the 10-, 50-, 100, and 200-word abstracts for current document set
3. Perdocs - single-document summaries for each document in current document set by a given assessor.

For this experiment the Perdocs are used as a combination of ideal summary 1 and ideal summary 2 against which the system generated summary is compared using the ROUGE-1.

The summaries in perdocs are abstract summaries generated by the assessors, the summaries are expected to be 100 words but there are cases where there are summaries with a slight difference, in which the summaries are greater or less than the 100 words restriction.

The automatic summary generated by the prototype is also restricted to extract only those sentences whose total count of words is a minimum of 100, i.e., if the first highly ranked sentences are four and their total word count is 93 then the system will also include the 5^{th} ranked sentence so that the minimum of the word count in extracted sentences is greater or equal to 100, but once the 100 word limit is reached additional sentences will not be considered.

Due to the problem of finding dedicated and professional summarizers the Amharic Ideal summaries are generated in different fashion. The English summaries are an abstract of the original document with hundred words regardless of the size of the document. Finding human summarizers, who can professionally, abstract each document with a limit of hundred words while covering all the topics and have two people to generate separate summaries for each document poses a logistical challenge.

Therefore the Amharic Ideal summaries were extracts of the document, where each summarizer extract the first highly ranked sentences whose word count is greater or equal to 100.

5.2 Summary Evaluation

For a smoother evaluation, establishing a reference summary and automating the process of evaluation becomes necessary. The usual practice in establishing the reference summary is to use human summarizers who either extract sentences or abstract based on a guideline, and the evaluation is automated using different algorithms. Among the different algorithms that can be used for automatic summary evaluation, ROUGE-1 is selected for this experiment due to

1. The fact that ROUGE-1 tends to favor a system generated summary which shares a measure of all the reference summaries, because this is intuitive and reasonable as we normally favor a candidate summary that is more similar to consensus among reference summaries [41].
2. The fact that previously done graph based summarization used ROUGE-1 for evaluating their summary performance.

A sample document, the system generated summary and the two human generated summaries are presented below for reference.

BC-Hurricane Gilbert 09-11 0339.

BC-Hurricane Gilbert,0348.

Hurricane Gilbert Heads Toward Dominican Coast.

By RUDDY GONZALEZ.

Associated Press Writer.

SANTO DOMINGO, Dominican Republic (AP).

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.

The storm was approaching from the southeast with sustained winds of 75 mph gusting to 92 mph.

"There is no need for alarm," Civil Defense Director Eugenio Cabral said in a television alert shortly before midnight Saturday.

Cabral said residents of the province of Barahona should closely follow Gilbert's movement. An estimated 100,000 people live in the province, including 70,000 in the city of Barahona, about 125 miles west of Santo Domingo .

Tropical Storm Gilbert formed in the eastern Caribbean and strengthened into a hurricane Saturday night.

The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.

The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm.

The weather service issued a flash flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast . There were no reports of casualties.

San Juan, on the north coast, had heavy rains and gusts Saturday, but they subsided during the night.

On Saturday, Hurricane Florence was downgraded to a tropical storm and its remnants pushed inland from the U.S. Gulf Coast.

Residents returned home, happy to find little damage from 80 mph winds and sheets of rain.

Florence, the sixth named storm of the 1988 Atlantic storm season, was the second hurricane.

The first, Debby, reached minimal hurricane strength briefly before hitting the Mexican coast last month.

Ideal Summary 1

Tropical Storm Gilbert in the eastern Caribbean strengthened into a hurricane Saturday night. The National Hurricane Center in Miami reported its position at 2 a.m. Sunday to be about 140 miles south of Puerto Rico and 200 miles southeast of Santo Domingo. It is moving westward at 15mph with a broad area of cloudiness and heavy weather with sustained winds of 75mph gusting to 92mph. The Dominican Republic's Civil Defense alerted that country's heavily populated south coast and the National Weather Service in San Juan, Puerto Rico issued a flood watch for Puerto Rico and the Virgin Islands until at least 6 p.m. Sunday.

Ideal Summary 2

Hurricane Gilbert is moving toward the Dominican Republic, where the residents of the south coast, especially the Barahona Province, have been alerted to prepare for heavy rains, and high winds and seas. Tropical Storm Gilbert formed in the eastern Caribbean and became a hurricane on Saturday night. By 2 a.m. Sunday it was about 200 miles southeast of Santo Domingo and moving westward at 15 mph with winds of 75 mph. Flooding is expected in Puerto Rico and the Virgin Islands. The second hurricane of the season, Florence, is now over the southern United States and downgraded to a tropical storm.

System Generated Summary

Hurricane Gilbert swept toward the Dominican Republic Sunday, and the Civil Defense alerted its heavily populated south coast to prepare for high winds, heavy rains and high seas.The National Hurricane Center in Miami reported its position at 2 a.m. Sunday at latitude 16.1 north, longitude 67.5 west, about 140 miles south of Ponce, Puerto Rico, and 200 miles southeast of Santo Domingo.The National Weather Service in San Juan, Puerto Rico, said Gilbert was moving westward at 15 mph with a "broad area of cloudiness and heavy weather" rotating around the center of the storm. Strong winds associated with the Gilbert brought coastal flooding, strong southeast winds and up to 12 feet feet to Puerto Rico's south coast.

Amharic Sample Data

ሲኖትራክ በሚል መጠሪያ የሚታወቀው ቻይና ሠራሽ የጭነት ተሽከርካሪ ምክንያት የሚከሰተው አደጋ እየተባባሰ በመምጣቱ፣ የአደጋውን መነሻ በመለየት ዕርምጃ ለመውሰድ መንግሥት ጥናት የሚያካሂድ ግብረ ኃይል ማቋቋሙ ተሰማ፡፡

የፌዴራል ትራንስፖርት ባለሥልጣን ምንጮች ለሪፖርተር እንደገለጹት፣ ባለሥልጣኑ ሲኖትራክ የተባለው ተሽከርካሪ በተለያዩ የአገሪቱ ክልሎች የሚያደርሰው አደጋ ከዕለት ወደ ዕለት እየጨመረ መሆኑን በመረዳቱ፣ ለአደጋው መንስዔ የተባሉትን ምክንያቶች የሚያጠና ቡድን አሰማርቷል፡፡ ቡድኑ ጥናቱን በአጭር ጊዜ አጠናቆ እንዲያቀርብ ተነግሮታል፡፡ ምንጮቹ ‹‹አስቸኳይ›› የተባለው ጥናት ምን ያህል ጊዜ እንደሚፈጅና ከጥናቱ በኋላ ስለሚወሰዱ ዕርምጃዎች ከመግለጽ ግን ተቆጥበዋል፡፡ ነገር ግን ቡድኑ በዋነኝነት የችግሮቹ መነሻ ከተሽከርካሪው የቴክኒክ ክፍሎች ወይስ ከአሽከርካሪዎች መሆኑን አጥንቶ ያቀርባል ሲሉ ገልጸዋል፡፡

ሪፖርተር ያነጋገራቸው የሲኖትራክ አስመጪዎችና አከፋፋዮች ችግሩን ከተሽከርካሪው የቴክኒክ ባህሪይ ጋር ያያይዙታል፡፡ ስማቸው እንዳይገለጽ የጠየቁ በአንድ አስመጪ ድርጅት የሚሠሩ መካኒክ ችግሩን ሲገልጹ፣ ‹‹ተሽከርካሪው የተሻለ ዘመናዊ ቴክኖሎጂ ቢኖረውም፣ የፍሬን መቆጣጠሪያው ችግር ያለበት በመሆኑ፣ የተሽከርካሪውን ፍጥነት ከክብደቱ አንፃር መቆጣጠር ስለማይቻል ብዙ ጊዜ ለአደጋ ተጋላጭ ነው፤›› ብለዋል፡፡ እንደ ባለሙያው ገለጻ የተሽከርካሪው ፍሬን በቶሎ የሚታዘዝ ባለመሆኑ፣ አሽከርካሪዎች ባህሪውን ቀድመው ካመረዳት የተነሳ መቆጣጠር ይሳናቸዋል፡፡

በሌላ በኩል አቶ ዘለዓለም ክፋይ የተባሉ የግል ተሽከርካሪ ባለቤትና አሽከርካሪ ግን፣ ችግሩ ከአሽከርካሪዎች የሥልጠና ማነስና ልማዳዊ የማሽከርከር ባህሪይ ጋር የሚያያዝ መሆኑን ይገልጻሉ፡፡ ‹‹መኪናዎቹ ዘመናዊና ከፍተኛ ፍጥነት ያላቸው ናቸው፡፡ ነገር ግን አሽከርካሪዎች ቀድሞ ባላቸው የማሽከርከር ልምድ ብቻ ስለሚያሽከረክሩና ለእንዲህ ዓይነት ተሽከርካሪ ምንም ዓይነት የተለየ ሥልጠና ስለማይወስዱ ለችግሩ ይጋለጣሉ፤›› በማለት ለሪፖርተር ገልጸዋል፡፡

የትራፊክ አደጋን በተመለከተ መረጃዎችን በመስጠት የሚታወቁት የአዲስ አበባ ፖሊስ ኮሚሽን የሕዝብ ግንኙነት ባለሙያ ረዳት ኢንስፔክተር አሰፋ መዝገቡ በበኩላቸው፣ አደጋው በአብዛኛው የሚደርሰው ከተሽከርካሪው የቴክኒክ ችግር ብቻ ሳይሆን፣ ከአሽከርካሪዎች ብቃት ማነስ መሆኑን የአብዛኞቹ የትራፊክ ፖሊሶች ሪፖርት እንደሚያመለክት አስታውቀዋል፡፡

ረዳት ኢስፔክተሩ ምንም እንኳ ወቅታዊ የትራፊክ አደጋዎችን በተለያዩ መገናኛ ብዙኃን በመስጠትና የግንዛቤ ትምህርቶችን በማሰራጨት ቢታወቁም፣ ራሳቸው በቅርቡ የሲኖትራክ አደጋ ሰለባ ሆነው እንደነበርና በአጋጣሚ መትረፋቸውን መዘገባችን አይዘነጋም፡፡

በሌላ በኩል ችግሩ ከተሽከርካሪዎችና ከአሽከርካሪዎች አልፎ እንደሚታይና ከገበያው ሥርዓት ጋር የሚያያዝ በመሆኑ፣ መንግሥት ጥናቱን ሰፋ አድርጎ ማየት እንዳለበትም የሚገልጹ የዘርፉ ባለሙያዎች አሉ፡፡ በተለይ ሪፖርተር ያነጋገራቸው ከተሽከርካሪው ዋና አከፋፋዮች አንዱ የሆነው አግታ ኃላፊነቱ የተወሰነ የግል ድርጅት ዋና ሥራ አስኪያጅ አቶ ግርማይ ገብረአረጋዊ፣ ‹‹ችግሩ በስፋት ሊታይ ይገባዋል፡፡ በተለይ መንግሥት ተሽከርካሪዎቹ ወደ አገር ውስጥ ስለሚገበቡበት ሁኔታ፣ የመለዋወጫ ዕቃዎች አገባብና አጠቃቀም፣ እንዲሁም በገበያው ውስጥ ያለውን አጠቃላይ አሠራር በጥንቃቄ ሊያየው ይገባል፤›› ሲሉ አሳስበዋል፡፡ ከዚህም ባሻገር ለተሽከርካሪው እዚህ አገር የሚከናወነው የመገጣጠም ሥራ ትክክል ነው ወይ ብሎ ሊያጣራም ይገባዋል ሲሉ የሚጠይቁት አቶ ግርማይ፣ በዚህ ዘርፍ ውስጥ ክፍተት እንዳለም ተናግረዋል፡፡

የትራፊክ አደጋ በየዕለቱ አሳሳቢ ደረጃ እየደረሰ መምጣቱን አብዛዎቹ የሚስማሙበት ሲሆን፣ ከቅርብ ጊዜ ወዲህ ግን በሲኖትራክ ተሽከርካሪ የሚደርሰው አደጋ እየከፋ መምጣቱን ሪፖርቶች ያሳያሉ፡፡ ችግሩ ከአዲስ አበባ አልፎም የክልል ቢሮዎችንም እያሳሰበ መሆኑን አንድ ስማቸው እንዳይጠቀስ የጠየቁ የፌዴራል ትራንስፖርት ባለሥልጣን ለሪፖርተር ገልጸዋል፡፡

እንደ ኃላፊው የካቲት 20 ቀን 2007 ዓ.ም. የክልሎች ትራንስፖርት ባለሥልጣን ኃላፊዎች በአዲስ አበባ ተገኝተው ሲኖትራክ የጭነት ተሽከርካሪ በየክልላቸው እያደረሰ ያለውን ጉዳት፣ ለባለሥልጣኑ ዋና ዳይሬክተር አቶ ካሳሁን ኃይለ ማርያም ገልጸዋል፡፡ በተለይ የኦሮሚያ ክልል ተወካይ በክልላቸው በአንድ ወር ውስጥ ደረሱ ያሉዋቸውን አራት አሰቃቂ አደጋዎች በማውሳት፣ የችግሩን አሳሳቢነት ማስረዳታቸውን ምንጮች ለሪፖርተር ገልጸዋል፡፡

ከሁለት ሳምንት በፊት 12 ሰዎችን አሳፍሮ ይጓዝ የነበረ ሚኒባስ ተሽከርካሪ ቡራዩ አካባቢ ከሲኖትራክ ጋር ተጋጭቶ ሚኒባሱ መሉ በሙሉ በመቃጠሉ፣ የ11 ተሳፋሪዎች ሕይወት ማለፉን መዘገቡ ይታወሳል፡፡ በተጨማሪም በዚሁ ዓመት ጎዳር ወር ከአዳማ ወደ አዋሽ ይጓዝ የነበረ አይሱዙ አውቶቡስ ከሲኖትራክ ጋር በመጋጨቱ ከ38 በላይ ተሳፋሪዎች መሞታቸው ለአብነት ይጠቀሳል፡፡

ባለሥልጣኑ ሲኖትራክ የተባለው ተሽከርካሪ በተለያዩ የአገሪቱ ክልሎች የሚያደርሰው አደጋ ከዕለት ወደ ዕለት እየጨመረ መሆኑን በመረዳቱ፣ ለአደጋው መንስዔ የተባሉትን ምክንያቶች የሚያጠና ቡድን አሰማርቷል፡፡ ነገር ግን ቡድኑ በዋነኝነት የችግሮቹ መነሻ ከተሽከርካሪው የቴክኒክ ክፍሎች ወይስ ከአሽከርካሪዎች መሆኑን አጥንቶ ያቀርባል ሲሉ ገልጸዋል፡፡ የትራፊክ አደጋን በተመለከተ መረጃዎችን በመስጠት የሚታወቁት የአዲስ አበባ ፖሊስ ኮሚሽን የሕዝብ ግንኙነት ባለሙያ ረዳት ኢንስፔክተር አሰፋ መዝገቡ በበኩላቸው፣አደጋው በአብዛኛው የሚደርሰው ከተሽከርካሪው የቴክኒክ ችግር ብቻ ሳይሆን፣ ከአሽከርካሪዎች ብቃት ማነስ መሆኑን የአብዛኞቹ የትራፊክ ፖሊሶች ሪፖርት እንደሚያመለክት አስታውቀዋል፡፡ በሌላ በኩል ችግሩ ከተሽከርካሪዎችና ከአሽከርካሪዎች አልፎ እንደሚታይና ከገበያው ሥርዓት ጋር የሚያያዝ በመሆኑ፣ መሥት ጥናቱን ሰፋ አድርጎ ማየት እንዳለበትም የሚገልጹ የዘርፉ ባለሙያዎች አሉ፡፡

Ideal Summary 1

ሲኖትራክ በሚል መጠሪያ የሚታወቀው ቻይና ሠራሽ የጭነት ተሽከርካሪ ምክንያት የሚከሰተው አደጋ እየተባባሰ በመምጣቱ፣ የአደጋውን መነሻ በመለየት ዕርምጃ ለመውሰድ መንግሥት ጥናት የሚያካሂድ ግብረ ኃይል ማቋቋሙ ተሰማ፡፡የትራፊክ አደጋን በተመለከተ መረጃዎችን በመስጠት የሚታወቁት የአዲስ አበባ ፖሊስ ኮሚሽን የሕዝብ ግንኙነት ባለሙያ ረዳት ኢንስፔክተር አሰፋ መዝገቡ በበኩላቸው፣ አደጋው በአብዛኛው የሚደርሰው ከተሽከርካሪው የቴክኒክ ችግር ብቻ ሳይሆን፣ ከአሽከርካሪዎች ብቃት ማነስ መሆኑን የአብዛኞቹ የትራፊክ ፖሊሶች ሪፖርት እንደሚያመለክት አስታውቀዋል፡፡ስማቸው እንዳይገለጽ የጠየቁ በአንድ አስመጪ ድርጅት የሚሠሩ መካኒክ ችግሩን ሲገልጹ፣ ‹‹ተሽከርካሪው የተሻለ ዘመናዊ ቴክኖሎጂ ቢኖረውም፣ የፍሬን መቆጣጠሪያው ችግር ያለበት በመሆኑ፣ የተሽከርካሪውን ፍጥነት ከክብደቱ አንፃር መቆጣጠር ስለማይቻል ብዙ ጊዜ ለአደጋ ተጋላጭ ነው፤›› ብለዋል።

Ideal Summary 2

የፌዴራል ትራንስፖርት ባለሥልጣን ምንጮች ለሪፖርተር እንደገለጹት፣ ባለሥልጣኑ ሲኖትራክ የተባለው ተሽከርካሪ በተለያዩ የአገሪቱ ክልሎች የሚያደርሰው አደጋ ከዕለት ወደ ዕለት እየጨመረ መሆኑን በመረዳቱ፣ ለአደጋው መንስዔ የተባሉትን ምክንያቶች የሚያጠና ቡድን አሰማርቷል፡፡ ነገር ግን ቡድኑ በዋነኝነት የችግሮቹ መነሻ ከተሽከርካሪው የቴክኒክ ከፍሎች ወይስ ከአሽከርካሪዎች መሆኑን አጥንቶ ያቀርባል ሲሉ ገልጸዋል፡፡ የትራፊክ አደጋን በተመለከተ መረጃዎችን በመስጠት የሚታወቁት የአዲስ አበባ ፖሊስ ኮሚሽን የሕዝብ ግንኙነት ባለሙያ ረዳት ኢንስፔክተር አሰፋ መዝገቡ በበኩላቸው፣አደጋው በአብዛኛው የሚደርሰው ከተሽከርካሪው የቴክኒክ ችግር ብቻ ሳይሆን፣ ከአሽከርካሪዎች ብቃት ማነስ መሆኑን የአብዛኞቹ የትራፊክ ፖሊሶች ሪፖርት እንደሚያመለክት አስታውቀዋል፡፡ በሌላ በኩል ችግሩ ከተሽከርካሪዎችና ከአሽከርካሪዎች አልፎ እንደሚታይና ከገበያው ሥርዓት ጋር የሚያያዝ በመሆኑ፣ መንግሥት ጥናቱን ሰፋ አድርጎ ማየት እንዳለበትም የሚገልጹ የዘርፉ ባለሙያዎች አሉ፡፡

System Generated Summary

የፌዴራል ትራንስፖርት ባለሥልጣን ምንጮች ለሪፖርተር እንደገለጹት፣ ባለሥልጣኑ ሲኖትራክ የተባለው ተሽከርካሪ በተለያዩ የአገሪቱ ክልሎች የሚያደርሰው አደጋ ከዕለት ወደ ዕለት እየጨመረ መሆኑን በመረዳቱ፣ ለአደጋው መንስዔ የተባሉትን ምክንያቶች የሚያጠና ቡድን አሰማርቷል።

የትራፊክ አደጋን በተመለከተ መረጃዎችን በመስጠት የሚታወቁት የአዲስ አበባ ፖሊስ ኮሚሽን የሕዝብ ግንኙነት ባለሙያ ረዳት ኢንስፔክተር አሰፋ መዝገቡ በበኩላቸው፣ አደጋው በአብዛኛው የሚደርሰው ከተሽከርካሪው የቴክኒክ ችግር ብቻ ሳይሆን፣ ከአሽከርካሪዎች ብቃት ማነስ መሆኑን የአብዛኞቹ የትራፊክ ፖሊሶች ሪፖርት እንደሚያመለክት አስታውቀዋል።

በሌላ በኩል አቶ ዘለዓለም ክፋይ የተባሉ የግል ተሽከርካሪ ባለቤትና አሽከርካሪ ግን፣ ችግሩ ከአሽከርካሪዎች የሥልጠና ማነስና ልማዳዊ የማሽከርከር ባህሪይ ጋር የሚያያዝ መሆኑን ይገልጻሉ።

ስማቸው እንዳይገለጽ የጠየቁ በአንድ አስመጪ ድርጅት የሚሠሩ መካኒክ ችግሩን ሲገልጹ፣ ‹‹ተሽከርካሪው የተሻለ ዘመናዊ ቴክኖሎጂ ቢኖረውም፣ የፍሬን መቆጣጠሪያው ችግር ያለበት በመሆኑ፣ የተሽከርካሪውን ፍጥነት ከክብደቱ አንፃር መቆጣጠር ስለማይቻል ብዙ ጊዜ ለአደጋ ተጋላጭ ነው፤›› ብለዋል።

5.2.1 ROUGE-N Result of Top Performing Systems in DUC 2002

Mihalcea and Tarau [13] reported the evaluation results of top five performing systems in the DUC 2002 single document summarization task which is shown in the following ***Table 5-1***:

Table 5-1 : DUC 2002 top performing Systems

Systems	ROUGE Score – ROUGE-1
	Stemmed and no stop words
S27	0.4405
S31	0.4160
S28	0.4346
S21	0.4222
S29	0.4019

In the same period TextRank reported a competitive result of ROUGE-1 result of 0.4229 (shown in Table 5-2) which is comparable with with the top performing systems shown in Table 5-1, the evaluation is done on DUC 2002 dataset.

Table 5-2 : TextRank Competitive Result after preprocessing

Systems	ROUGE Score – ROUGE-1
	Stemmed and no stop words
TextRank	0.4229

On the same DUC 2002 dataset our new algorithm have shown a remarkable improvement over the top 5 performing systems S27,S31, S28, S21 and S29 whose ROUGE-1 result is shown in Table 5-1. Our new algorithm, as well, outperforms TextRank whose ROUGE-1 result is shown in Table 5-2 with an average ROUGE-1 of 0.4229.

5.2.2 ROUGE-N Result of our proposed Algorithm

Our new Algorithm shows an average ROUGE-1 result of 0.5238 on the 264 documents from DUC 2002 document set as presented in Table 5-3.

Table 5-3 : Proposed Algorithm ROUGE Result - English

Systems – For English	ROUGE Score – ROUGE-1
	Stemmed and no stop words
Proposed New Alogrithm	0.5238

The evalution on the Amharic document set reports a greater ROUGE-1 result with 0.04% improvement, but the Amharic reference summaries are extracts instead of human abstracts unlike the DUC perdoc summaries.

Table 5-4 : Proposed Algorithm ROUGE Result – Amharic

Systems – For Amharic	ROUGE Score – ROUGE-1
	Stemmed and no stop words
Proposed New Alogrithm	0.5644

5.3 Discussion

The intituitive explanation to graph based extractive summarization is that a sentence is important in a document if it is connected to more important sentences in that document, in addition to the number of sentences that it is connected to. For example a sentence which is connected with 10 other sentences, which inturn are not connected with other sentences should be less favored than a sentence which is connected with other 6 sentences which are connected with other six sentences.

As stated earlier such ranking of nodes can be achieved by using Google's PageRank but we argue that adoption of PageRank for extractive text summarization has the following two issues:

1. In all the available means of computing PageRank its computation is iterative, where the number of iterations depends on a complex set of variables such as the size of the document (number of sentences), the convergence threshold selected and the weight between edges (content similarity).
2. When calculating the rank of pages using the iterative method, PageRank normalizes the rank of inbound pages by the number of their outbound links, this reduction of a page's rank for referencing other pages can't be maintained in ranking sentences because there will be no logical explanation for normalizing the rank of a sentence for referencing (sharing a content) with other sentences.

When we say that PageRank is iterative we mean that each page is ranked a number of times before the difference between two consequative computation of a Page's rank are below a given threshold, in the case of TextRank the threshold being 0.0001.

Our new algorithm ranks and extract sentences without iteration, this is possible because we can determine the importance of each sentence in a document in advance by considering the weight of its similarities with the other sentences incident upon it and the degree of the sentence.

This predetermination of a page's standing is impossible in a graph of web links because in web link's a page's importance is dependent on the importance of page's which are referecing it, and the referencing page's importance is in turn dependent upon on the importance of those page's referencing it and this chain continues in the entire network, since it is impossible to determine a page's importance (how probable it is to reach at that page starting from any random page in the network) by simply considering the number of inlinks and outlinks in that page.

This is because ultimately PageRank is the embodiement of the random surfer model which models the probability of a web page being accessed starting from any given random page by continuous click without clicking back.

Therefore our algorithm which takes advantage of the weight of edges between sentences nodes in combination with the sentences degree can succinctly rank the standing of a sentence in a document independently, then the sentence rank which considers the independent rank of sentences in the document rank sentences gives a better result and avoid the iteration that are inherent in PageRank.

Chapter 6
Conclusion and Future Works

6.1 Conclusion

If automatic summarization is intended to facilitate information retrieval and consumption, no part of the process should be manual. Not only should it avoid manual intervention but the summarization has to be generic, not affected by the genre of the document.

Even though there are a dozen of algorithms in automating text summarization ranging from using a common sense approach of using word distributions upto the use sophisticated learning algorithm, most of them are restricted by their requirement of parameterization or genre specific features.

Yet the use of graph theory in extractive summarization is completely unsupervised and generic means of generating automatic summarization. The entire process of summarization can be done without any human involvement, without any machine learning or parameterization, without considering document specific features like position of sentences.

Graph theoretic approach has been used both in single and multi document summarization, and has reported a comparable result with supervised and genre specific approaches.

This thesis has taken graph based automatic extractive text summarization a notch up by introducing a new sentence ranking approach. Previous graph based approach used the degree based summarization which considers all sentences to be of equal importance or adopt Google's PageRank which ranks sentences by considering "prestigious" sentences more favorably than "less prestigious" ones or in other words ranking sentences both the number of their connections and by the prestige (importance) of those sentences to which they are connected.

This thesis has contributed a new algorithm of ranking sentences based on both the number of their connections (degree) and the prestige (importance) of sentences they are connected to, the new algorithm is better than the adoption of Google's PageRank because

1. It reduces the computational complexity of PageRank which is in polynomial order represented by the big **O** notation O (n^c) where starts with a minimum of 4 and goes to a conditional value dependent upon the the number of sentences, the convergence value

selected and the weight of similarity between nodes to a maximum of quadratic order complexity represented by the big **O** notation O (n^2) where each sentence is ranked just once.

2. The new algorithm resulted in a better informative summary which is evidenced by the improved ROUGE-1 result as shown in subsection 5.2.1.

6.2 Future Works

Almost all solutions to a problem come with their advantage and disadvantage and automatic text summarization is no exceptions, even though compared to other summarization algorithms graph based approach is entirely unsupervised and generic it has got its own limitations.

Among the limitations the development of an exhaustive list of stop words and building a robust stemming package is the major challenges we faced while working on this thesis.

Other limitations we observed is that the summary generated tends to focus on the major topic by deemphasizing sub topics.

To that end, we forsee the following three potential enhancements for future work:

a. using tf*idf with a given thresholding to prune sentences into a bag of content words before building the similarity matrix and ranking sentences, as this will avoid the need for the use of stop words list and stemming, to use edit distance on the bag of words instead of text similarity measures based on content overlap, yet the outcome of this approach is to be verified by the researcher.
b. To consider a paragraph based summarization, where sentences will be ranked first in their paragraph and then to form a new document by picking all the sentences which are ranked first in their respective paragraph, and then to rank those sentences anew or to do a variant of this, instead of considering only sentences ranked first, to consider all ranks. This approach is intended to generate a well representative informative summary which includes not only the main topic, but also the subtopics
c. With the use of tf*idf and thresholding as suggested in “a” to prune sentences into a bag of content words it is very much viable to extend our new algorithm to generate a multi document summary.

References

[1] H. Hovy, "Automated Text Summarization," in *The Oxford Handbook of Computational Linguistics*, Oxford, Oxford University Press, 2005, pp. 583 - 598.

[2] V.Gupta,S.Lehal, "A Survey of Text Summarization Extractive Techniques," *Journal of Emerging Technologies In Web Intelligence,* pp. 258 - 268, 2010.

[3] P.Luhn, "The Automatic Creation of Literature Abstracts," *IBM Journal,* pp. 159-165, 1958.

[4] P.Edmundson, "New Methods in Automatic Extracting," *Journal of the ACTM,* pp. 264 - 285, 1969.

[5] G.PadmaPriya,K.Duraiswamy, "An Approach for Concept-based Automatic Multi-Document Summarization," *International Journal of Applied Information Systems,* vol. 3, no. 2249-0868, pp. 49-53, 2012.

[6] T. Hoffman, "Probabilistic Latent Semantic Analysis," *Uncertainity In Artificial Intelligence,* 1999.

[7] G.Erkan, D.R. Radev, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artifical Intelligence Research ,* vol. 22, pp. 457-479, 2004.

[8] Habtamu Demlie, *Multi-Document Amharic Text Summarization Using Probabilistic Latent Semantic Analysis (PLSA),* Addis Ababa: Addis Ababa University, 2014.

[9] Melese Tamiru, *Automatic Amharic Text Summarization using Latent Semantic Analysis,* Msc Thesis, Addis Ababa University, 2009.

[10] Eyob Delele, *Topic-based Amharic Text Summarization,* Addis Ababa: Addis Ababa University, 2011.

[11] T. F. P. &. L. D. Landauer, "Introduction To Latent Semantic Analysis," 1998, pp. 259-284.

[12] D. Oneata, "Probablistic Latent Semantic Analysis".

[13] R.Mihalcea, P. Tarau, "TextRank: Bringing Order into Texts.," in *EMNLP*, Barcelona, 2004.

[14] Y.Lin, "ROUGE Working Note," 20 January 2015. [Online]. Available: http://research.microsoft.com/en-us/um/people/cyl/download/papers/rouge-working-note-v1.3.1.pdf.

[15] T. Visser, B. Wieling, "Sentence-based Summarization of Scientific Documents," [Online]. Available: http://www.martijnwieling.nl/files/wielingvisser05automaticsummarization.pdf. [Accessed 13 01 2015].

[16] A. Gupta, M.Joshi, P. Drungarwal, "Document Summarization," 10 October 2014. [Online]. Available: www.cse.iitb.ac.in/~pb/cs626-sem1-2012/.../summarization-oct12.pptx.

[17] "Automatic Summarization," 21 January 2015. [Online]. Available: http://en.wikipedia.org/wiki/Automatic_summarization.

[18] "Content Word," 21 January 2015. [Online]. Available: http://en.wikipedia.org/wiki/Content_word.

[19] "What is Discourse," 21 January 2015. [Online]. Available: http://www-01.sil.org/linguistics/glossaryoflinguisticterms/WhatIsADiscourse.htm.

[20] "TFIDF," 23 January 2015. [Online]. Available: http://www.tfidf.com/.

[21] L.Neto, A. Freitas, A. Keistner, "Automatic Text Summarization using Machine Learning Approach," [Online]. Available: http://www.cs.kent.ac.uk/people/staff/aaf/pub_papers.dir/SBIA-2002-Joel.pdf. [Accessed 12 May 15].

[22] K.Ganesan,C.Zhai, J.Han, "Opinosis : A Graph-Based Approach to Abstractive Summarization of Highly Redundant," in *23rd International Conference on Computational Linguistic.* , Beijing, 2010.

[23] D.Y.Sahhare, R.Kumar, "http://www.mecs-press.org/," 05 03 2014. [Online]. Available: http://www.mecs-press.org/ijitcs/ijitcs-v6-n3/IJITCS-V6-N3-5.pdf. [Accessed 23 02 2015].

[24] E. Lloret, "Text Summarization : An Overview," 23 January 2015. [Online]. Available: www.dlsi.ua.es/~elloret/.../TextSummarization.pdf.

[25] V. Steen, An Introduction to Graph Theory and Complex Networks, 2010.

[26] "An Introductory Course on Network Analysis," Google, 29 July 2010. [Online]. Available: https://sites.google.com/site/networkanalysisacourse/schedule/an-introduction-to-centrality-measures. [Accessed 22 April 2015].

[27] S. Brin,L. Page, "The Anatomy of Large Scale Hyper Textual Web Search Engine," Stanford University, Stanford, 1998.

[28] R.Tansae, R.Radu, "The Mathematics of Web Search," Cornell University, 2009. [Online]. Available: http://www.math.cornell.edu/~mec/Winter2009/RalucaRemus/index.html. [Accessed 24 April 2015].

[29] J. Steinberger, K. Jezek, "Evaluation Measure For Text Summarization," *Computing and Informatics,* pp. 1001-1026, 2009.

[30] A. Huang, "Similarity Measure For Text Document Clustering," University of Waikato, Hamilton.

[31] H. Saggion, D. Radev, S.Teufel, W. Lam, S.M.Strassel, "Developing Infrastructure for the Evaluation of Single and Multi-Document Summarization Systems in a Cross-Lingual Environment," 23 January 2015. [Online].

[32] Y. Lin, *ROUGE Working Note,* Southern California: Information Sience Institute, University of California, 2004.

[33] J.Dorr, C.Monz, S.President, R.Schwarts, D.Zajic, "A Methodology for Extrinsic Evaluation of Text Sumamrization : Does ROUGE Correlate?," in *ACL Worshop on Intrinsic and Extrinsic Evaluation Measures For Machine Translation and/or Summarization*, Ann Arbor, 2005.

[34] W.Isenberg, Grammar of The Amharic Language, Charleston SC: Nabu Press, 1842.

[35] R.Nordquist, "Punctuation," [Online]. Available: http://grammar.about.com/od/pq/g/punctuationterm.htm. [Accessed 28 January 2015].

[36] "Amharic," [Online]. Available: http://en.wikipedia.org/wiki/Amharic. [Accessed 28 January 2015].

[37] Nega Alemayehu,P.Willet, "Stemming of Amharic Words For Information Retrieval," Litrary and Linguistic Computing, Shiefield, 2002.

[38] Y.Gong, X.Liu, "Generic Text Summarization Using Relevance Measure and Latent Semantic Analysis," [Online]. Available: http://www.cs.bham.ac.uk/~pxt/IDA/text_summary.pdf . [Accessed 26 January 2015].

[39] H.Bhandari, M.Shimbo, T.Ito and Y.Matsumoto , "Generic Text Summarization using Probabilistic Latent Semantic Indexing," in *Asian Federation of Natural Language Processing*, Hyderabad, India, 2008.

[40] A. Andres, "http://combine.it.lth.se," 21 04 2006. [Online]. Available: http://combine.it.lth.se/CrawlSim/report/node34.html. [Accessed 28 June 15].

[41] Y. Lin, "ROUGE: A Package for Automatic Evaluation of Summaries," in *ACL*, Barcelona, 2004.

Annex – C# Code of Sentence Rank

C# Code for SentenceRank

```
public void SentenceRank()
        {

            double SentR = 0;
            try
            {
                for (int i = 0; i < removeEmptySentences.Count; i++)
                {
                    SentR = 0;
                    for (int j = 0; j < removeEmptySentences.Count; j++)
                    {
                        if (i + 3 != j)
                     SentR += Convert.ToDouble(SentenceMatrix[j + 3, i]) *
                     Convert.ToDouble(SentenceMatrix[2, i]);
                    }
                    //Simply the sum of (SentIR * CurrentOverlap) for
                    //overlapping sentences
                    SentenceMatrix[1, i] = SentR.ToString();
                }
            }
            catch (FormatException ex)
            {

                string msg = ex.Message;
            }

        }
```

public void DocumentGraphBuilder()

```
        {
            List<string> sentences = new List<string>();
            /*
              *Computes the number of Content Overlaps of the sentences in
              CopySentenceMatrix List
             *and assign the corresponding value in the SentencesMatrix
             *ACTUALLY THIS DOES
             *1. COMPUTE SIMILARITY BETWEEN SENTENCES
             *2. COMPUTE INDEPENDENT RANK
             */

            //index (i,0),(i,1),(i,2) will contain the following values always
            //0 - the sentence
            //1 - Sentence Rank
            //2 - the Independent Rank

            int countOverlaps = 0;
```

```
        int contentOverlap = 0;

        for (int k = 0; k < removeEmptySentences.Count; k++)
        {
            sentences.Add(CopySentenceMatrix[0, k]);
        }

        for (int i = 0; i < sentences.Count; i++)
        {

            //SentenceMatrix[0, i] = sentences[i];

            for (int j = 0; j < sentences.Count; j++)
            {
                //don't calculate similarity for the same sentences
                if (i != j)
                {
                 //get the content overlap to count sentences overlapping with the
                 sentence
                    contentOverlap = CalculateContentOverlap(sentences[i],
                    sentences[j]);
                    if (contentOverlap > 0)
                    {

                      countOverlaps = countOverlaps + contentOverlap;
                    }

                    SentenceMatrix[j + 3, i] = CalculateContentOverlap(sentences[i],
                    sentences[j]).ToString();
                }

            }
            //for updating the out count

            SentenceMatrix[2, i] = countOverlaps.ToString();
            //reset for the next sentence
            countOverlaps = 0;
        }

    }
```

*where Sentence Matrix and remove empty sentences are global variables

Decleration

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.

Declared By:

Name: _____

Signature: _____

Date: _____

Confirmed By Advisor:

Name: _____

Signature: _____

Date: _____

Place and Date of Submission: **Addis Ababa, Ethiopia. June 2015.**