Addis Ababa University

College of Natural Sciences

Modeling an Automatic Amharic Text Summarizer: Abstractive Approach

Mohammed Abdella Hassen

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

September, 2016

Addis Ababa University

College of Natural Sciences

Modeling an Automatic Amharic Text Summarizer: Abstractive Approach

Mohammed Abdella Hassen

Advisor: Yaregal Assabie (PhD)

This is to certify that the thesis prepared by Mohammed Abdella, titled: *Modeling an Automatic Amharic Text Summarizer: Abstractive Approach* and submitted in partial fulfillment of the requirements for the Degree of Master of Science in Computer Science complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the Examining Committee:

	Name	Signature	Date
Advisor:	Yaregal Assabie (PhD)	_____	
Examiner:	(PhD)	_____	
Examiner:	(PhD)	_____	

Abstract

The need for automatic text summarization systems increase as the number of electronic documents that deal with specific information increases in the web. The two basic approaches of text summarization systems are extractive and abstractive. Extractive approach is based on selecting the most important sentences from the input document using different algorithms and presents the selected sentences as a summary for the input document. The abstractive approach for text summarization tries to generate novel sentences that may not be present in the input document but still represent the main idea of the input document. The abstractive approach is based on the semantic representation of input sentences.

This thesis proposes an automatic Amharic text summarizer using abstractive approach based on the Universal Networking Language (UNL) which is one of the semantic representations of natural language sentences. We use different components that are related with UNL representation. Related sentences in the input document are clustered and each cluster will have its own generated sentence to be used as a summary. Thus, the number of summary sentences is based on the number of clusters formed from the input document. The text preprocessing stage which involves processes like normalization, stop-word removal and stemming makes the input data suitable for clustering component by giving the root forms or stems from the relevant words of an input sentence. The conversion between the natural language sentence and the UNL expression are done using the EnConversion or DeConversion rules together with the morphological properties of each of the words in an input sentence. There is also another component which is UNL analysis that is used for providing the common UNL expression from a group of UNL expressions.

In order to evaluate the performance of the proposed system, we use Amharic input documents and human evaluators that are going to evaluate based on different parameters. The parameters used to evaluate the performance of the system are the grammar of the summary sentences and the idea represented in the summary. The results of the evaluation are promising since we use the subjective evaluation of summary sentences.

Key words: text summarization, Universal Networking Language (UNL), EnConversion, DeConversion

Acknowledgment

I would like to take this opportunity to express my heart-felt gratitude for those who helped me throughout my journey. I would like to thank the almighty Allah for making all things possible. Next, my gratitude goes to my advisor Yaregal Assabie (PhD) for his encouragement, valuable suggestions and comments that helped me in doing the research work.

I would like to thank my families for their support not only in this research but throughout my education journey. I would also like to thank my colleagues and friends for their encouragement and support.

1 Contents

List of Figures

List of Tables

Acronyms and Abbreviations

NLP	Natural Language Processing
UNL	Universal Networking Language
UW	Universal Word

Chapter One: Introduction

1.1 Background

Nowadays, the data available on the web is growing rapidly and users access this data in different ways. Because the amount of documents published in a specific field is too large, it is difficult for individuals to read all documents in a specific field. As a result, document summarization methods become the subject of many researches in the area of information retrieval [1].

Text summary can be 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) [2]. The summary conveys to the reader the main ideas of the document and consequently the reader can determine whether the complete document is of any relevance or not.

Text summarization is the process of distilling the most important information from a source (or sources) to produce an abridged version for a particular user (or users) and task (or tasks) [3]. It is a tool that can help users to go through that vast volume of data available. Text summarization can be conducted through human professionals of the specific domain but this is time consuming and costly. Hence, this calls for the need of automatic text summarization [4]. When this process is done by means of a computer, i.e. automatically, we call it automatic text summarization [5].

Automatic text summarization is the process of automatically creating a compressed version of a given text that provides useful information to the reader. Automatic text summarization can also be defined as the creation of a briefer representation of a body of information by a computer program. The product of this procedure should still contain the most central facts of the original information. Thus, automatic text summarization, analogously, is the shortening of texts by computer while still retaining the most important points of the original text [6].

Text summarization systems can give generic summary of a given text or a summary based on a user query. If it gives a summary based on a user query, it is called query-driven summarization. If it gives a summary by keeping all relevant features of the original text, we call it generic.

There are two types of summaries based on their purpose to the user and these are informative and indicative. An indicative summary gives an idea on what the document is about. It gives abbreviated information by preserving the most important portions of the original document. Indicative summaries are often returned by search engines as a response for user queries. On the other hand, informative summary tries to give as much information as possible to the user and to be used as a substitute for the original document. The typical length of an indicative summary is from 5-10% of the full document whereas that of informative summary is from 20-30% [7].

Generally there are two approaches for automatic text summarization. These are extractive and abstractive approaches to summarize texts. In extractive approach of text summarization, the summary is extracted by selecting representative sentences or phrases from the original text based on some weight. But in abstractive approach of text summarization, the summary contains word sequences that may not be present in the original text [7]. Abstractive approach of text summarization is very challenging since it involves understanding the semantics of the original text and generating novel sentences that can be used as a summary.

The process of automatic text summarization has three stages. These are topic identification, interpretation and generation [2]. When the approach for text summarization is extraction, it does the first stage only. This stage involves determining the most important units such as words, paragraphs or sentences from the original text. The units will be given a score depending on the algorithm applied and sentences that contain highest scoring units will be used as a summary for the original document.

Abstraction approach for text summarization undergoes the remaining two phases after identifying the most important units from the original text. The second stage which is interpretation involves representing the important units identified in stage one with words or concepts that may not be found in the original text. Different units may be fused and represented using a new concept based on the semantic understanding of the original text.

The last stage in automatic text summarization is generation and it involves generation of valid sentences from the interpreted representation of important units that can be found in stage two. Here in this step the sentences to be generated are novel sentences and a single sentence may be a representative of many sentences in the original text [2, 8].

Text summarization by abstraction is more challenging task than that of extraction since it requires parsing the original text in a deep linguistic way, interpreting the text semantically into a formal representation, finding new more concise concepts to describe the text and generating a new shorter text that have the same information content as the original text [6].

In the last stage of automatic text summarization by abstraction, i.e., generation, it needs analyzing of words, phrases and sentences of a language since it is required for generating valid sentences. The grammar should also be modeled to understand the grammar of the input text and to determine the correct grammar for the sentence(s) to be generated as a summary.

The simple and most used method of selecting sentences to be included in the summary is using the frequency of words [3]. It assumes that the frequency of a word in a text is directly proportional to the relevance of the word to the text. It sorts words in decreasing order of their frequency. It then calculates the relevance score of each sentence based on the frequency score of words they contain. The best scoring sentences will be included in the summary.

Automatic text summarization systems for Amharic language have been done by various researchers. From these researches we can mention the works done by Melesse Tamiru [4] and

Habtamu Demile [9]. These works are good but they are restricted to give sentences that are present in the original text which means they follow extractive approach of text summarization. Both researchers do not try to generate novel sentences that may not be present in the original text.

Automatic text summarization has many applications in the area of information retrieval and information extraction. Automatic summaries can be displayed as a search result for the user to be used in decision making. Digital libraries and journals may use automatic summaries of journals to be displayed for their users before the user decides to go through the whole document. Search engines may use automatic summaries to index the whole document and this improves the performance of the search engines.

1.2 Motivation

The motivation for doing a research work on Amharic text summarization system that follow abstractive approach is the lack of research work done on Amharic text summarization systems that follow abstractive approach. Many researches are done regarding Amharic text summarization, but, to the best knowledge of the researcher, there is no research work done on text summarization systems using abstractive approach. So, these motivate us to do a research on this area.

1.3 Statement of the problem

Automatic text summarization for different languages has been done so far by different researchers. There are also few research works that aim at developing an automatic text summarizer for Amharic language. But, to the best knowledge of the researcher, all of the works for Amharic language follow an extractive approach which is a process of selecting representative sentences from the original text.

In an extractive text summarization approach, some sentences may be included in the summary simply because they constitute a single word which is relevant. But in abstractive approach, that relevant word will be used together with other relevant words from other sentences and generate a novel sentence. This will reduce the size of the summary to be produced.

There is no research work for Amharic language in developing an automatic text summarizer that can generate novel sentences. If the summarizer generates novel sentences that can be a representative of the original text, it will be an effective summary since it can represent the whole text without being restricted to existing sentences only.

So, these motivate us to develop an automatic Amharic text summarizer that can generate novel sentences that are going to be used as a summary of the original text.

1.4 Objectives

General Objective

The general objective of this research is to develop abstractive text summarizer for Amharic language.

Specific Objectives

The following specific objectives are identified in order to achieve the specified general objective:

- Reviewing literatures and analyze automatic text summarization methods in general and automatic summarization methods that follow abstraction approach in particular
- Collecting Amharic text corpus
- Grammar modeling for Amharic language
- Developing Amharic word, phrase and sentence analyzer
- Developing Amharic word, phrase and sentence generator
- Designing a model for abstractive Amharic text summarizer
- Developing a prototype for the summarizer
- Testing the performance of the prototype

1.5 Methodology

The following methods are applied in order to achieve the above specified objectives.

Literature Review

A thorough literature review is done on text summarization methods in general and abstractive approach of text summarization in particular with regard to techniques used in each approach. Stages for abstractive text summarization are also investigated and techniques or tools that can be used as a component for abstractive text summarization are adopted.

Data Collection

The data to be used for text corpus is obtained from various sources of Amharic text that are available on the web. The collected data is distributed to human evaluators for evaluating the system summaries using different criteria. Unlike evaluation methods for text summarization systems that are based on extractive approach, the evaluators are not expected to generate manual summaries. Rather, they used other evaluation criteria like the grammar of the summary, whether the main idea of the original document is presented in the summary or not.

Tools

To accomplish the study, Java programming and python are used. We select java because it is easy to build applications and it supports Unicode encoding to be used for Amharic language. Python is selected as there are components of the system that are partially developed with and we use some of the features from them.

Prototype Development

In order to evaluate the performance of the methods we will propose, a prototype system will be developed for the Amharic text summarizer that can generate novel sentences. The performance of the prototype will be evaluated by language experts.

1.6 Scope and Limitations

The components of the proposed system are designed to consider simple sentences of Amharic language. The grammar correctness of the input sentences is not handled in the system. The main focus of our system is the applicability of simple rules for different processes in the proposed system.

1.7 Application of Results

The contribution of this thesis work is to lay the foundation for Amharic text summarization systems that follow abstractive approach. It uses different components that are useful for natural language applications that are on Amharic language. The research work will also have a big contribution in the developing of full-fledged Amharic text summarizer. It could be used as an important component for natural language applications that involve Amharic language.

1.8 Organization of the Rest of the Thesis

The rest of this thesis report is organized as follows. Chapter two will discuss about different issues on text summarization systems. It will have information about what text summarization means, different types of text summarization systems and different approaches to text summarization systems. It will also discuss on the grammar rules and morphological rules of Amharic language and the semantic representation of natural language sentences using UNL expression.

Chapter three will review different works done on text summarization in general and particularly for Amharic language and will try to see the gaps in previous text summarization systems done for Amharic language. Chapter four will present the proposed system for Amharic text summarizer and details about different components of the proposed system. Chapter five will present the experiment done to evaluate the performance of the proposed system and in chapter six; we will provide the conclusion and the future research directions related to text summarization systems.

Chapter Two: Literature Review

2.1 Introduction

This chapter gives a brief discussion about the field of automatic text summarization. There are different types of text summaries depending on different factors. These types of summaries will be discussed in this chapter. We also investigate the important stages in performing text summarization and the methods for evaluation of text summaries. The different approaches to text summarization that are found in the literature will also be discussed.

The other main point to be discussed in this chapter is modeling the grammar of a language as it is a vital component of text summarization systems that are based on abstraction.

2.2 Types of Text Summarization

There are three factors that will be used for classification of text summaries. These are input factor, output factor and purpose. Depending on the number of input texts, we can categorize text summarization systems as single document and multi-document summarization. When we use the output as a factor of classification, we can get extractive and abstractive summaries. And if we use the purpose of the summary, we may get indicative and informative summaries or query-based and generic summarization systems.

2.2.1 Extractive vs. Abstractive Summarization

The main difference between extractive and abstractive methods of summarization is in their output. Extractive summarization methods simply select relevant information to be included in the summary from the original text and give as a summary without modifying the original text. Abstractive summarization systems give a summary that may not be existed in the original text. Hence, abstractive summarization systems are more challenging since it requires parsing the original text in a deep linguistic way, interpreting the text semantically into a formal representation, finding new more concise concepts to describe the text and generating a new shorter text that have the same information content as the original text[6].

The other challenge in text summarization systems that follow abstractive approach is modeling the grammar of the language. Modeling the grammar is important to understand the input text and to give grammatically valid output. A single sentence from the summary may have the ideas of many sentences from the original text. Hence, it may need fusion of concepts that are related and may be found in different sentences of the original text.

2.2.2 Single Document vs. Multi-document Summarization

The number of documents used as an input for summarization systems decide which method of summarization to be used. If the input is a single document, we call it single document

summarization system. And if the summary is obtained from multiple documents, it is called multi-document summarization. Multi-document summarization is more difficult than single document summarization since it involves extra processes like avoiding redundancy and ordering of sentences [9].

Avoiding redundancy is not an issue in single document summarization since a single document rarely repeats itself but multiple documents that are on the same area may contain similar sentences. Thus, the summary should avoid redundant sentences. Ordering of sentences is also an issue in multi-document summarization because the order of sentences in the summary should be coherent and determining the order of sentences from different documents is a challenging task but is not an issue in single document summarization.

2.2.3 Indicative vs. Informative

Based on the content covered by the summary, we can classify text summarization systems as informative or indicative. An indicative summary gives an idea on what the document is about. It gives abbreviated information by preserving the most important portions of the original document. Indicative summaries are often returned by search engines as a response for user queries. On the other hand, informative summary tries to give as much information as possible to the user and to be used as a substitute for the original document. The typical length of an indicative summary is from 5-10% of the full document whereas that of informative summary is from 20-30% [7].

2.2.4 Query-driven vs. Generic

The other classification based on the purpose of the summary is classifying text summarization systems as generic or query-driven. Generic summary tries to give a summary based on the whole information found in the original document. Query-driven summaries contain information based on the user's query. It will accept a user query and searches for information related to that query in the document and tries to summarize the related information.

2.3 Stages of Automatic Text Summarization

The process of automatic text summarization has three stages. These are topic identification, interpretation and generation [2]. When the approach for text summarization is extraction, it does the first stage only. Topic identification involves determining the most important units such as words, paragraphs or sentences from the original text. The units will be given a score depending on the algorithm applied and sentences that contain highest scoring units will be used as a summary for the original document.

Topic identification usually starts with document pre-processing where the document undergoes several processes to be represented by terms capable of representing the content of the document

and then applies several summarization techniques to select important units of a document [10, 4].

Text preprocessing

Applications that involve the process of natural languages usually have a component that performs the preprocessing task on the input text. Text preprocessing is the most important step in the area of computational linguistics, since the quality of the summary depends on how efficient is the presentation of a text [11]. The tasks involved in text preprocessing are described below.

Segmentation

Documents are composed of units like words, phrases, sentences and paragraphs. The smallest unit that will be extracted from the original document for the summary should be decided before performing summarization. The smallest unit can be a paragraph, a sentence, a phrase or a word. In order to find out the boundaries of the phrases/sentences/words, the input text is segmented into tokens. Segmentation is not a simple task since there are irregularities in natural languages. For most of the languages, white spaces and punctuation marks are used as boundary markers [12]. In Amharic language, sentences are identified by the delimiter "። ". Words are identified by means of spaces and a number of Amharic punctuation marks such as ፡ , ፤ and ! [13].

Normalization

Normalization is performed on the tokens that are resulted from the segmentation stage. Normalization is the process changing different forms of a given text to a single form that will be a representative for all forms throughout the process. In Amharic, there are two issues of normalization [14].

The first one is identification of Amharic alphabets that have the same use and pronunciation but different representation. These alphabets should be replaced by one of the alphabets that are chosen as a representative from the set of alphabets which have the same use and pronunciation. For example, the words ሰላይ and ሠላይ have the same meaning, usage and pronunciation but different representation. Thus, they have to be changed to ሰላይ.

The second issue of normalization is the short forms of a word using the slash "/" or the period ".". For example, the word ት/ቤት should be normalized to the word ትምህርት ቤት.

Stop Words Removal

Stop words are low information bearing words such as "and" or "the", typically appearing with highest frequency. Stop words may be selected as relevant words because of the high occurrence but they will not have relevant meaning. High frequency words have higher variance and effective weight in many methods, causing them to be erroneously selected as features due to sample noise. To deal with these nuisance words, many text processing methods use a fixed,

hand-built stop-word list and preemptively remove all features on that list from consideration [15].

Stemming

Stemming is the process of converting words to their stem (root) form. In documents, a word can be seen in different formats, such as plural vs. singular, present vs. past tense, etc. Most of the time these words have the same meaning and treating them differently is unnecessary. In order to use these words as the same token (concept), stemmers are used. Stemmers are necessary to represent different word forms in a single format and to reduce memory usage for storing the words. As a result of performing stemming, document representation (input matrix) is less noisy and denser [12]. Using stemmers also have an advantage in the performance of the text processing system since different words but with the same stem will be considered as a single word and this result in better precision and recall of the system.

The second and third stages of automatic text summarization are done by systems that follow abstractive approach. The second stage involves interpreting the extracted portion of a document that is considered as salient depending on the technique or approach used. The third stage generates the summary using generating techniques from the representation of salient information in stage two.

The problem with extractive approach is it falls short of producing optimal summaries in terms of content and linguistic quality. In contrast to most automatic text summarization systems, people tend to produce abstractive summaries by rewriting unclear phrases and paraphrasing to produce a concise version of the content found in the input document. Peoples also re-use portions of an input document but by cutting pieces of a document and combining with portion of another part of the document [16].

The other problem of extractive summary is its content may inadvertently include unnecessary detail along with salient information. This is because once an extractive approach determines that a sentence in an input document contains salient information, all information in that sentence will be included in the summary regardless of the relevance of other part of the sentence. Text summarization systems that follow an abstractive approach attempt to extract salient information from sentences and combine with other information to produce a novel sentence. The basic idea of novel sentences may be a collection of the idea of many sentences in the input document.

2.4 Approaches to Text Summarization

There are lots of approaches to text summarization systems in the literature. Some of the approaches to text summarization are discussed in this section.

2.4.1 Surface Level Approach

Most of the early systems of automatic text summarization follow the Surface level approach. This approach inclines to use shallow features of a text and selectively combining them together in order to obtain a salience function that can be used to extract information [5]. The first work done on automatic text summarization is that of Luhn and follows the surface level approach. It assumes that the frequency of words is directly proportional to the relevance of that word to the whole document [3].

The first step proposed by Luhn for text summarization after preprocessing is counting the frequency of words and putting the words in descending order. Sentences will be given score based on the occurrence of those relevant words. The highest scoring sentences will be included in the summary.

The other factor used in surface level approaches is the location of words. Usually sentences that come first have more relevance and should be included in the summary but this is genre-dependent. The other method assumes that the importance of sentences is determined by the presence of words from the title, headings or initial part of a text [3].

Another example of surface level approach is using cue words. Cue words or phrases are used to determine or signal the relevance or irrelevance of a sentence to the whole document. In general there are three categories of cue phrases: bonus, stigma, and null phrases. Bonus phrases are used to emphasize the importance of a sentence in a text while stigma phrases reflect that the sentence is not important. Null phrases are neutral phrases and are not considered when the weight of a sentence is computed. Few examples of bonus phrases are „significantly", „in conclusion", „in this paper we show", etc. whereas „hardly" and „impossible" are examples of stigma phrases. Thus, each cue phrase is assigned a positive or negative relevance. The weight of each sentence is then the sum of the weights of the words in it [4].

2.4.2 Entity Level Approach

Entity level approach attempts to build a representation of the text by modeling text entities and their relationships. This will be an indication of the connectivity of entities that determine salient information. There are various relationship types between words like similarity, proximity, co-occurrence, co-reference, logical relations, syntactic relations and repetition [5].

Similarity of words occurs when two words share a common stem i.e. whose form is similar. It can also be calculated by vocabulary overlap or using linguistic techniques. This similarity level can be extended to phrase or paragraphs. Based on the similarity measure, it is possible to build relationship between entities and the degree of connectedness will be used to give the score for the relevance of the sentence.

Proximity refers to the distance between text units. Using the degree of proximity, it is possible to establish entity relationships that can be used as criteria for determining scores of relevance.

Co-occurrence is also a measure that determines the relation formed by occurring in common texts.

The other methods of measuring similarity between entities are logical relation between entities such as agreement, contradiction, entailment and consistency. We can also use the syntactic relation which is based on parse trees to measure the similarity of words or entities.

2.4.3 Discourse Level Approach

Discourse deals with the properties of the text as a whole that convey meaning by making connections between component sentences. Discourse refers to any form of language-based communication involving multiple sentences or utterances such as text and dialog [17].

The target of discourse level approaches is to model the global structure of the text and its relations in order to achieve communicative goals. The information that can be exploited at discourse level includes format of the document such as hypertext markup or document outlines, threads of topics as they are revealed in the text and rhetorical structure of text that represent argumentative or narrative structure. The idea behind rhetorical structure of text is to deal with the possibility of building the coherence structure of a text, so that the centrality of textual units will reflect their importance [5].

2.4.4 Machine Learning Approach

Machine learning approach is an approach used by training a program to identify summary sentences using an existing text/summary corpus. As there are more and more indicators of sentence importance to be included in the summary, it becomes necessary to come up a method that combines the different indicators. Machine learning techniques on a corpus of document/summary pairs is claimed that statistical analysis of the corpus would reveal what features should be used and how they should be weighted relative to each other [18]. Some examples of machine learning techniques are using Naïve Bayes classifier and Hidden Markov Model (HMM). The Naïve Bayes classifier is based on the assumption that the employed features used to classify sentences as summary-sentences or not are independent of each other. The HMM, which has fewer independence assumptions than Naïve Bayes, is based on the hidden dependency to be identified using the corpus. There is a result from HMM that shows the probability of the next sentence in a document to be included in the summary will depend on whether the current document sentence is part of the summary [19].

Summary sentences in the training corpus were matched against their corresponding document using a variety of techniques, such as exact match, join of two article sentences for the abstract sentences or incomplete partial matches, fixed or cue phrases, paragraph features related to the position of sentences and word frequency. The problem inherent in the supervised learning paradigm is the necessity of labeled data on which classifiers can be trained. Asking annotators to select summary-worthy sentences is time consuming. So that many researchers have

concentrated their efforts on developing methods for automatic alignment of human abstracts and the input document in order to provide labeled data of summary and non-summary sentences for machine learning [20]. However, this approach may be problematic since different writers can choose different content for their summary and therefore summary-worthy sentences may not be identifiable based on a single abstract. To overcome this problem, it is proposed to leverage the information from manual evaluation of content selection in summarization in which multiple sentences can be marked as expressing the same fact that should be in the summary [21].

2.4.5 Sentence Compression Approach

Sentences of a summary produced by summarization systems that follow extractive approach contain unnecessary information along with useful facts. This is because sentences that contain salient information will be included without modifying the original sentence. If the original sentence is not modified, unnecessary information that is contained in the sentence will be present in the summary [16]. There is evidence on the corpus analysis of document summaries produced by human professional summarizers. The corpus is prepared by Jing and McKeown [22] and they observed that sentence reduction was often used by professional summarizers. For their analysis of human professional summarizing behavior, they worked on the corpus which is a collection of newspaper articles about computer products along with a human summary of each article.

The analysis of the behavior of human summarizers is studied by automatically aligning sentences in the abstracts to the sentences in the original article and studying the sentence-level transformation employed by the writers of the abstracts. They found that 78% of the summary sentences were written by editing the input document and of those, more than half of the edits were done using sentence compression alone, removing information from a sentence extracted from the input document. The remaining edits used compression in addition to combining information from one or more other sentences. Thus, compression is an important component of summarization systems that follow an abstractive approach.

There are two general approaches to sentence compression [23]. The first one is rule-based approach that uses primarily linguistic techniques and the second one is statistical approach which is based on statistical techniques. Rule-based approach to sentence compression uses both syntactic and discourse knowledge to determine how to compress a sentence. Syntactic knowledge includes the syntactic structure of an extracted sentence and knowledge about which constituents are less likely to be needed, while discourse knowledge includes information about how each constituent is connected with the rest of the summary.

Statistical approach to sentence compression uses a mechanism to learn about the rules of which syntactic constituents are going to be deleted. In this approach, no linguistic rules that show how to determine a sentence need to be provided. The system is going to learn from a corpus of compressed sentence and original long sentences how the compression is done.

2.4.6 Information Fusion Approach

The main idea behind information fusion is human abstractors, in addition to sentence compression, sometimes substitute one word for another and often combine information from two sentences to create a novel sentence. This was noticed by extensive analysis of the types of edits that are typically carried by human abstractors and observed just one operator other than sentence compression. This operator is used to create a new sentence using conjunction between two or more reduced sentences from the input text under certain constraints such as having the same information [24].

A more general approach to information fusion from several sentences is used for multi-document summarization. It deals with finding similarities across the input documents and extracting the similarities to form the summary. Often similarities are identified using sentence clustering but a text-to-text generation technique that takes as input asset of similar sentences and produces a new sentence containing information common to most of the input sentences. This approach addresses two challenges: the identification of phrases that convey common information and the combination of those phrases in to a novel grammatical sentence [25].

2.5 Summary Evaluation

Evaluation is the final stage in any research and the key indicator of the applicability of the results especially in cases where there are many methods and approaches of doing same tasks and producing different results. In text summarization, it needs to be evaluated to determine how well the summary can be used for the required purpose as there are many approaches and techniques for the summarization process [4].

For text summarization, there is no standard method of evaluation but different researchers use their own method of evaluation. Thus, there may be different methods of evaluation depending on the need of the researcher. But, generally we can classify evaluation systems for text summarization as intrinsic and extrinsic [26].

2.5.1 Intrinsic Evaluation

Intrinsic evaluation, also known as normative evaluation, is an assessment of the quality of a summary based on text quality. Text quality refers to grammaticality, non-redundancy, reference clarity and coherence of the summary. It may be involved by human judges to determine the quality of the summary or by comparing the summary with the ideal summary written by the author of the source text [27].

There are different methods of intrinsic evaluation. One of such methods is using co-selection measures. Co-selection measure is used to evaluate summaries using the metrics of precision, recall and F-score. The calculation of these metrics is based on the number of sentences that can be found from the system summary and the ideal summary.

Precision (p) is the number of sentences occurring in both the system and ideal or human summaries divided by the number of sentences in the system summary. Recall (r) is the number of sentences occurring in both the system and ideal or human summaries divided by the number of sentences in the ideal or human summary. F-score is a composite measure that combines precision and recall. The basic way how to compute the f-score is to count a harmonic average of precision and recall [28]. These measures are calculated using the following equations

$$P(Precision) = \frac{system\ and\ human\ summary\ overlap}{system\ summary}$$

$$R(Recall) = \frac{system\ and\ human\ summary\ overlap}{human\ summary}$$

$$F\ (F\text{-}score) = \frac{2 * P * R}{P + R}$$

The advantage of using co-selection method is once the human judges define the ideal summary, it can be used to evaluate automatic summaries with a simple comparison. But, precision might be too strict in that some of the sentences chosen by the system might be good though they have not been chosen by the ideal summary. Thus, giving more emphasis to recall than precision is suggested as recall measures the overlap with already observed sentences.

The other method of evaluation is content-based measures of evaluation. It tries to overcome the drawback of the co-selection measure as it matches only exactly the same sentences by ignoring the fact that two sentences may contain the same information even if they are written differently.

The best example of content-based measures of evaluation is cosine similarity. Cosine similarity uses the vector space model to represent both the system and ideal summaries. The words that are found in both summaries will be used as the axes of the vector space. Hence, both the system and ideal summaries can be represented as points in the vector space. We can measure the similarity of the two summaries by taking the cosine of the angle between the vectors that represent the two summaries. The angle between the two vectors is inversely proportional to the similarity between the two summaries.

Other approach of intrinsic approach is using language experts to judge whether the summary is good or not. Language experts will assess the produced summary and decide on it. This approach of evaluation is expensive since it needs human judges and the subjectivity of judging with human being may be biased.

The main problem in intrinsic approach of evaluation is the difficulty of constructing the ideal summary for a given text. Even if we can construct the ideal summary, since it is constructed by

human beings, different human beings will have different ideal summaries. So, selecting the ideal summary is a difficult task.

2.5.2 Extrinsic Evaluation

Extrinsic or task-based evaluation aims at measuring the performance of using the summaries for a certain task. It differs from intrinsic evaluation as it doesn't check the quality of text rather it checks the performance of some other tasks that use the summaries as an input. For example we can assess how well the summary helps the user to determine the source's relevance to topics of interest.

The main drawback of extrinsic evaluation is its difficulty as it requires other fully functional systems or tasks that take the produced summary as an input. It is also time-consuming, expensive and requires considerable amount of careful planning [29].

2.6 The Amharic Language

Amharic is one of the major languages spoken in Ethiopia. Although many languages are spoken in Ethiopia, Amharic is the dominant language as it is a mother tongue for a substantial segment of the population and it is the most commonly learned second language throughout the country [30]. The language is a member of the Semitic branch of the Afro-Asiatic language family. It is the official working language of the federal government of Ethiopia and some of the regional governments such as Amhara, Gambella, Benishangul Gumuz and South people's regional states etc. it is also spoken by more than 25 million peoples throughout the country.

2.6.1 Amharic Morphology

Amharic is one of the morphologically rich languages. Like other Semitic languages, Amharic exhibits a root-pattern morphological phenomenon. A root is a set of consonants (also called radicals) which has a basic lexical meaning. A stem in Amharic language is formed by inserting vowels or pattern on vowels in to the consonants of a root. This is the process of non-concatenative morphological feature. In addition to this, Amharic uses different affixes to create inflectional and derivational word forms [31].

There are five parts of speech in Amharic which are Adjectives, nouns, verbs, adverbs and prepositions and conjunctions. Prepositions and conjunctions are totally unproductive. Adverbs are few in number and are less productive. They are not inflected but some adverbs can be derived from adjectives. For instance, the adverb "ክፉኛ" which means "severely or seriously" is derived from the adjective "ክፉ" which means "wicked". It is done by suffixing "-ኛ" to the adjective. Thus, our focus of morphology will be on the remaining three parts of speech, i.e. nouns, verbs and adjectives.

Nouns are derived from other basic nouns, adjectives, stems, roots and the infinitive form of a verb by affixation and intercalation. Case, number, definiteness and gender marker affixes inflect the noun. Adverbs can be derived from adjectives.

Adjectives in Amharic are derived from nouns, stems or verbal roots by adding a prefix or a suffix. Moreover adjectives can also be formed through compounding. Adjectives are inflected for gender, number and case.

Amharic verbs are derived from roots. The conversion from a root to a verb stem is done by intercalation and affixation. For instance, from the root word "ግድል" kill, we obtain the perfective verb stem "ገደለ" by intercalating the pattern c1ኧc2c2ኧc3ኧ where c1, c2 and c3 are the set of consonants or radicals and "ኧ" is the vowel in Amharic language. From this perfective stem, it is possible to derive a passive "ተገደለ" and a causative stem "አስገደለ" using prefixes "ተ-" and "አስ-" respectively. Other verb forms are also derived from roots in a similar fashion. Verbs are inflected for person, gender, number, aspect, tense and mood [32]. Other elements like negative markers also inflect verbs in Amharic.

Here we have to discuss briefly the morphological process of Amharic language. As we discuss earlier, as Amharic is one of Semitic languages, it follows a root-pattern morphological phenomenon. In addition to root-pattern, it has also a derivational and inflectional morphological process.

Root-pattern Morphology

Root-pattern morphology is a characteristic of Semitic languages. As we have seen earlier, a root or radical is a set of consonants which carry the basic lexical meaning of a word. The number of radicals in a word is commonly three but ranges from one to six. A pattern consists of vowels which are inserted or intercalated among the consonants of a word. The pattern is combined with a particular affixes and create a single grammatical form or another stem. Stems are formed by intercalating the vowels among root consonants. Affixes are added to the stem in order to form another stem or to complete the stem to be a word [32, 33].

The root-pattern morphology of Amharic language is the non concatenative one. For example, the Amharic root "ስብር" means "break", when we insert the pattern CኧCCኧC among the radicals, we get the stem ሰበር. Attaching the suffix -ኧ gives ሰበረ "he broke" which is the first form of the verb (third person masculine singular in past tense). Using the same pattern, but without geminating the second consonant i.e. the pattern CኧCኧC makes and attaching the suffix –ኣ, we get the process nominal "ሰበራ" which means "breaking". The pattern CኧCኣC combined with the suffix –ኢ makes agent nouns like "ሰባሪ" which means "one who breaks". The same stem with the suffix –ኣ, gives us the participle "ሰባራ" which means "broken". Intercalating the vowel a between the second and third consonant, i.e. the pattern CCኧC, results in a jussive stem "ስበር". Attaching the prefix መ- to it forms an infinitive verb "መስበር" which means "to break". The process of forming stems using a particular pattern is indicated by a sequence of Cs and Vs,

which is called a template, where C stands for consonants and V for vowels. For example, CVCCVC, CCVC and CVCC are templates that represent the perfective, jussive and imperfective stems of tri-radical verbs respectively.

Derivational Morphology

Nouns can be derived from other nouns, adjectives, roots, stems and the infinitive form of a verb by affixation and intercalation. The morphemes **-ነት**, -ኛኛ, -ኛት, -ኣዊ, -ተኛ, -ኛ and the prefix ባለ- are used to derive nouns from other nouns. Table 2.1 shows examples of nouns that are derived from other nouns.

Table 2.1 nouns derived from other nouns

Base noun	**Meaning**	**Bound morpheme**	**Derived noun**	**Meaning**
ልጅ	child	-ነት	ልጅነት	Childhood
ግንብ	wall	-ኛኛ	ግንበኛ	One who build a wall
ሹም	Appointed	-ኛት	ሹመት	Appointment
ኢትዮጵያ	Ethiopia	-ኣዊ	ኢትዮጵያዊ	Ethiopian
ድንበር	Border	-ተኛ	ድንበርተኛ	One who shares a border
እንግሊዝ	England	-ኛ	እንግሊዝኛ	English
ቤት	House	ባለ-	ባለቤት	House owner

Nouns can also be derived from adjectives using the suffixes **-ነት** and -ኛት as in the examples ደግነት ‘generosity’ which is derived from the adjective ደግ and ዕውቀት ‘knowledge’ from the adjective ዕውቅ ‘known’.

The other method of deriving a noun is from verbal roots by intercalation and affixation. One possibility to derive a noun from a root is intercalating the vowel እ among the root consonants or just after the first root consonant. This intercalation may also result in a bound morpheme which,

together with different affixes, used to form other nouns. Intercalation of the vowel ኧ after the first radical or among radicals' results in either a noun or a bound stem used to derive a noun. The pattern ኧ - ኣ and the suffix - ኢ are used in the derivation of agent nouns. Nouns of manner can be derived by prefixing አ - to the stem which is formed by duplicating the penultimate radical and intercalating the pattern ኧ - ኣ - ኧ. The infinitive/ verbal noun is derived by prefixing the morpheme መ - to the jussive verb stem and the instrumental noun is derived by suffixing - ኢያ to the infinitive. Table 2.2 shows examples of nouns that ate derived from verbal roots.

Table 2.2 nouns derived from verbal roots

Root	Stem	Affix	Derived noun	Meaning
ል - ብ - ስ	ልእብስ		ልብስ	Cloth
ግ - ር - ድ	ግእርድ	- ኦሽ	ግርዶሽ	Eclipse
ስ - ን - ፍ	ስእንፍ	- ና	ስንፍና	Laziness
ድ - ግ - ም	ድእግግእም	- ኦሽ	ድግግሞሽ	Repetition
ስ - ብ - ር	ስእብር	- ኣት	ስብራት	Breakage
ስ - ር - ቅ	ስእርቅ	- ኦት	ስርቆት	Theft
ድ - ክ - ም	ድእክኣም		ድካም	Tiredness
ት - ክ - ዝ	ትእክክኣዝ	- ኤ	ትካዜ	Sadness
ግ - ፍ	ግእፍ	- ኢያ	ግፊያ	Crush
ግ - ፍ	ግእፍ	- ኢት	ግፊት	Influence
ሽ - ፍ - ት	ሽእፍት	- ኣ	ሽፍታ	Bandit
ቅ - ል - ድ	ቅኧልድ		ቀልድ	Joke

ቅ - ል - ም	ቅኧልኧም		ቀለም	Color, ink
ስ - ብ - ር	ስኧብኧር	- ኣ	ሰበራ	Process of breaking
ስ - ብ - ክ	ስኧብኣክ	- ኢ	ሰባኪ	Preacher
ው - ድ - ቅ	ውእድድኣቅ	- ኢ	ውዳቂ	Rubbish
ስ - ብ - ር	ስብኧር	መ -	መስበር	To break
ስ - ብ - ር	ስብኧር	መ - , - ኢያ	መስበሪያ	Tool of breaking
ስ - ብ - ር	ስስኧብኣብኧር	አ -	አሰባበር	Manner of breaking

Adjectives in Amharic include the words that modify nouns and can be modified by the word በጣም ‘very, greatly’. Adjectives are derived from nouns, stems or verbal roots by adding a suffix and by intercalation. The suffixes ኣም ኛኛ ኣዊ ኣማ are used in the derivation of adjectives from nouns. For example, it is possible to derive ሀብታም ‘rich, wealthy’, ሀይለኛ ‘powerful, mighty’, ዘመናዊ ‘modern’ and ድንጋያማ ‘stony’ from the noun ሀብት ‘riches, wealth’, ሀይል ‘power, force’, ዘመን ‘period, epoch’ and ድንጋይ ‘stone’ respectively.

Adjectives can also be derived from roots by intercalation of vocalic elements or attaching a suffix to bound stems. The pattern ኧ - ኣ will produce a bound stem used to produce adjectives by adding the suffix - ኣ. For example, from the root ጥ - ም - ም, intercalating the pattern ኧ - ኣ produce the bound stem ጠማም. Adding the suffix - ኣ will give the adjective ጠማማ.

The most complex morphology occur in Amharic is the morphology of verbs. In Amharic verbs, the most important element is the penultimate radical. Traditionally, Amharic verbs are classified in to three depending on the gemination pattern of the penultimate radical. In type A verbs, the penultimate radical geminates in perfect tense only. In type B verbs, the penultimate radical geminates irrespective of the verb forms. In type C verbs, the penultimate radical geminates in both perfect and imperfect verb forms. Table 2.3 gives an example of different tenses of verbs for the three classifications of verbs. More detail derivation of verbs from [32].

Table 2.3 nouns derived from verbal roots

Verb forms	Type A: ሰ - ብ - ር		Type B: ፍ - ል - ግ		Type C: ም - ር - ክ	
	Stems	Template	Stems	Template	Stems	Template
Perfect	ሰ ኧ ብ ብ ኧ ር	CVCCVC	ፍ ኧ ል ል ኧ ግ	CVCCVC	ም ኣ ር ር ኧ ክ	CVCCVC
Imperfect	ሰ ኧ ብ ር	CVCC	ፍ ኧ ል ል ግ	CVCCC	ም ኣ ር ክ ክ	CVCCC
Jussive	ሰ ብ ኧ ር	CCVC	ፍ ኧ ል ል ግ	CVCCC	ም ኣ ር ክ	CVCC
Gerund	ሰ ኧ ብ ር	CVCC	ፍ ኧ ል ል ግ	CVCCC	ም ኣ ር ክ	CVCC
Infinitive	ሰ ብ ኧ ር	CCVC	ፍ ኧ ል ል ኧ ግ	CVCCVC	ም ኣ ር ኧ ክ	CVCVC

Inflectional Morphology

Amharic nouns inflect for case, number, definiteness and gender marker affixes. In Amharic, there are three cases; nominative, accusative and genitive. Nominative has no indicator and it is distinguished by its place in a sentence where nominative comes always before accusative [30, 32]. The suffixes - ን, - ዬ, - ህ, - ሽ, - ዎ, - ው, - ዋ, - ኣቸው, - ኣችን, - ኣችሁ etc. inflect Amharic nouns for genitive case whereas the prefix የ- inflects Amharic nouns for accusative.

Amharic has only two numbers, singular and plural. The suffixes are used to inflect nouns for plural number. For example, ቄሶች ‘priests’, ህፃናት ‘babies’ and ሰባኪያን ‘preachers’ are derived from nouns ቄስ ‘priest’, ህፃን ‘baby’ and ሰባኪ ‘preacher’ respectively. There are some exception words which that use vocalic changes and reduplication to indicate plurality like ደናግል ‘virgins’ which is derived from the word ድንግል ‘virgin’.

A noun in Amharic can be either definite or indefinite. Indefiniteness is marked using the numeral አንድ ‘one’. For instance, the phrase አንድ ቤት ‘one house’ or ‘a house’ depending on the context indicates the house is indefinite. The definite markers are the bound morphemes ኡ and ው for masculine nouns ending with consonant and vowels respectively. The bound morphemes - ዋ, - ኢቱ, - ይቱ, -ኢትዋ and - ይትዋ can be used for feminine nouns. For plural nouns, we use the morpheme ኡ which is attached to the noun after the plural marker. If a definite noun is preceded by an adjective, the definiteness marker is attached to the adjective instead of the noun. For example, the phrase ትልቅ ቤት has two words ትልቅ adjective and ቤት noun. When we attach the definite marker - ኡ, we will attach to the adjective instead of the word which gives the phrase ትልቁ ቤት which is a definite noun phrase.

Amharic has only two genders, masculine and feminine, and distinguishes gender in the second and third person. Masculine nouns do not have gender marker morphemes whereas to indicate feminine nouns, we use morphemes - ኢት and - ኢቱ. Some words do not indicate whether they refer to a masculine or feminine like the word ሙሽራ. In this case, we use morphemes - ው and - ዋ to give words ሙሽራው and ሙሽራዋ for masculine and feminine respectively. In some cases, the demonstrative pronoun ይህ - and ይህች - refer the nouns being used are masculine and feminine respectively.

Adjectives inflect for case, number, definiteness and gender in a similar fashion to nouns. For example the adjective ትልቅ can be inflected as ትልቁ and ትልቋ by adding suffixes - ኡ and - ዋ for gender. It can also be inflected as የትልቁ for case, gender and definiteness whereas ትላልቅ for plural.

Verbs in Amharic are inflected for person, gender, number, aspect, tense and mood [32]. Table 2.3 shows an example inflection of the verbal root ሰ - በ - ር for person, gender and number of the perfective verb.

Table 2.4 subject marker of a verb

Person		Perfective verb
1st	Singular	ሰበር ኩሁ
	Plural	ሰበር ን
2nd	Masculine	ሰበር ክህ
	Feminine	ሰበር ሽ
	Polite	ሰበር ኡ
	Plural	ሰበር ኣችሁ
3rd	Masculine	ሰበር ኧ
	Feminine	ሰበር ኧች
	Polite	ሰበር ኡ
	Plural	ሰበር ኡ

With regard to aspect, Amharic verbs are categorized in to two classes, perfect and imperfect forms. The imperfective aspect is indicated by the prefixes ል - and ይ - before the stem which is followed by the auxiliary verbs ነው and ነበር. The perfective aspect is expressed using the prefix እየ - attached to a perfective stem and with the auxiliary verbs ነው and ነበር.

In Amharic, there are four types of moods which are declarative, interrogative, negative and imperative. Verbs can take different forms according to the mood and this can be expressed either in the stem or by inflectional affixes. Table 2.4 shows the structure of inflection of the verbal root ስ - ብ - ር according to the mood.

Table 2.5 inflection of verbs according to mood

Person		Declarative	Interrogative	Negative	Imperative
1st	Singular	እ - ሰብር -	ል - እ - ስበር -	አ - ል - ስበር	
	Plural	እ - ን - ሰብር -	እ - ን - ስበር -	አ - ን - ስበር	
2nd	Masculine	ት - ሰብር -	ት - ሰብር -	አ - ት - ስበር	ስበር -
	Feminine	ት - ሰብር - ኢ	ት - ሰብር - ኢ	አ - ት - ስበር - ኢ	ስበር - ኢ
	Polite	ት - ሰብር - ኡ	ት - ሰብር - ኡ	አ - ት - ስበር - ኡ	ስበር - ኡ
	Plural	ት - ሰብር - ኡ	ት - ሰብር - ኡ	አ - ት - ስበር - ኡ	ስበር - ኡ
3rd	Masculine	ይ - ሰብር -	ይ - ስበር -	አ - ይ - ስበር -	ይ - ስበር -
	Feminine	ት - ሰብር -	ት - ስበር -	አ - ት - ስበር -	ት - ስበር -
	Polite	ይ - ሰብር - ኡ	ይ - ስበር - ኡ	አ - ይ - ስበር - ኡ	ይ - ስበር - ኡ
	Plural	ይ - ሰብር - ኡ	ይ - ስበር - ኡ	አ - ይ - ስበር - ኡ	ይ - ስበር - ኡ

Tense in Amharic can be classified as past and non-past. The past tense can further be categorized as simple past, recent past and remote past. Simple past is referred by a perfective stem and shows that an action is completed in the past but does not indicate the exact time unless adverbs of time are used. For example, the sentence አበበ መጽሃፍ ገዛ ‘Abebe bought a book’ indicates the past tense but does not indicate the actual time when Abebe bought a book whereas አበበ ትናንት መጽሃፍ ገዛ ‘Abebe bought a book yesterday’ clearly indicates the time of the action. Recent past and remote past are formed by using the completive aspect stem of the verb together

with the suffix - ኣል and the auxiliary verb ነበር ‘was’ respectively. For example, አበበ መጽሃፍ ገዝቷል ‘Abebe bought a book’ will be recent past and አበበ መጽሃፍ ገዝቶ ነበር ‘Abebe bought a book’ will be remote past.

The time of non-past tense in Amharic ranges from present to future. It is formed by the imperfective stem with a bound morpheme - ኣል attached to the verb. The stem indicates the action is not performed and the bound morpheme indicates the action will be performed in some time. For example, the sentence አበበ መጽሃፍ ይገዛል ‘Abebe will buy a book’ is non-past and the time of the action can range from present to future. It is ambiguous to know it is present tense or future unless adverbs of time are used like the sentence አበበ ነገ መጽሃፍ ይገዛል ‘Abebe will buy a book tomorrow’ or አበበ በሚቀጥለው ሳምንት መጽሃፍ ይገዛል ‘Abebe will buy a book next week’ indicate the exact time of the action.

2.6.2 Amharic Grammar

This section discusses about the syntax or grammar of Amharic sentences. The grammar of a sentence in a language deals with the arrangement of words or word categories in a sentence and the agreement between this word categories. We discuss some of the concepts of word arrangement and agreement of word categories in a sentence but detail rules with examples can be found in [32].

Arrangement

Depending on the language, words of a sentence should be in their correct order to convey the desired meaning. Amharic has a subject-object-verb (SOV) arrangement. However, sometimes the sentences could have an order of object-subject-verb (OSV). But, this arrangement is used in informal texts. For example, in the sentence አበበ መጽሃፍ ገዛ ‘Abebe bought a book’ አበበ ‘Abebe’ is a noun which is the subject of the sentence and መጽሃፍ ‘a book’ is also a noun which is the object of the sentence whereas ገዛ ‘bought’ is a verb.

Adjectives and adverbs are modifiers that elaborate or give a clear picture of the thing they modify. Adjectives are used to modify nouns whereas adverbs are used to modify verbs. In Amharic language, adjectives appear before the noun they modify. For example, in a phrase ረጅም ዛፍ ‘tall tree’, the adjective ረጅም ‘tall’ comes before the noun it modifies which is ዛፍ ‘tree’. Adverbs come before the verb they modify. For example, in the sentence አበበ ነገ መጽሃፍ ይገዛል ‘Abebe will come tomorrow’, the adverb ነገ ‘tomorrow’ comes before the verb it modifies which is ይገዛል ‘will come’.

Agreement

Agreement between the lexical categories of words in a sentence depending on different situations should be achieved to have grammatically correct sentences. The first agreement type is subject-verb agreement. Verbs should agree with their subject in number, gender and person.

For example, the sentences አበበ በሶ በላች is grammatically incorrect since the subject አበበ is a masculine noun and the verb በላች is inflected to indicate a feminine noun by the suffix -ች. The verb of a sentence should also agree with the objects that precede verbs with in the verb phrase. For example, the sentence is grammatically አመሃ ዶሮውን አረዳት is grammatically incorrect since the object ዶሮውን is inflected for masculine noun by using the suffix – ው and the verb አረዳት indicates that the object is a feminine. So, the verb should be inflected by using the suffix – ው and should be አረደው to indicate masculine noun and make the sentence correct.

The other agreement is the agreement between the adjective and the noun that the adjective modifies. Adjectives which inflect the number and gender of a noun that it modifies, the adjective should agree with the number and gender of a noun to be modified. For example, the phrase is ትልቅ ቤት incorrect since the adjective ትልቅ refers to a singular noun whereas the noun ቤት is plural. The phrase ትልቁ ሴትዮ is also incorrect since the adjective ትልቁ refers a masculine noun but the noun ሴትዮ is a feminine.

Another agreement of lexical categories of words is between the adverb and verb. Since the adverbs are used to modify the verb they precede, they have to agree with the verb they modify. For example, the sentence አበበ ነገ መጣ is grammatically incorrect since the adverb ነገ indicates the time which is in future whereas the verb መጣ indicates that the action is already performed in the past. So, the verb መጣ should be changed to a verb form which indicates the future time like ይመጣል to make it a correct sentence using the prefix ይ – and the suffix - ኣል.

2.7 Semantic representation

Semantic representation of natural language sentences is needed in text summarization systems that follow abstractive approach. It is needed to represent the meanings of the original sentences and provide new representation of a natural language sentences that can be considered as equivalent or related in meaning to a group of original sentences. In our research, we use the Universal Networking Language (UNL) to represent the meaning of input sentences. In this section, we discuss the process of representing the semantics of a natural language using UNL.

In our research work, as it is a summarization system, we use the UNL representation of sentences to analyze and come up with a representation of sentences which has the central idea of many related sentences. Since the original idea and usage of UNL if for machine translation systems, it involves two different languages. But in our case, since there is one natural language involved, we make a change in the structure of the UNL component. It uses morphological analyzer and generator to construct a UNL representation and to generate natural language sentences from UNL representation.

What is UNL?

UNL is a computer language that enables computers to process information and knowledge. It is used to replicate the function of natural language. It can be used to describe all information and knowledge conveyed by natural languages for computers. This results for peoples to have a linguistic infrastructure in computers to understand multi-lingual information.

The purpose of introducing the Universal Networking Language (UNL) in representing semantic meaning of a natural language is to achieve accurate exchange of information among different languages and representing a solution to overcome the barriers of linguistic differences. The main aim of UNL is to have the same representation for sentences in different languages that are the same in meaning. It acts as an intermediate representation in machine translation (MT) systems [35].

The core softwares in the UNL framework are the EnConverter and the DeConverter. The software tool used to convert a source language (natural language) expression into the UNL expression is referred to as EnConverter and the software tool used to convert UNL expressions into a target language representation is called DeConverter. The two software tools use a set of grammar rules and a word dictionary of target language to perform the process of converting from natural language to UNL representation and vice versa. But, in our case, as there is a single natural language, there is no need to have a dictionary of target language. We use this software tools to perform the EnConversion and DeConversion process for a single language.

UNL Expression

The UNL uses semantic network to represent natural languages. This semantic network is a directed graph. Its nodes are UWs (or "scope" as it is commonly called) representing concepts and edges are Relations between concepts. Concepts can be annotated by Attributes. Such a semantic network of the UNL is called a "UNL Expression" or "UNL Graph". A UNL expression is a binary relation consists of the three semantic units which are called Universal Words (UWs), relations and attributes.

Universal Words

UWs are words of the UNL, constitute the UNL vocabulary. UWs are used as labels for concepts, syntactic and semantic units to form a UNL expression. A combination of a set of UWs, linked with each other through relations and modified by attributes - expresses the meaning of a sentence. A UW in UNL expression is defined in the following format:

<UW> =:: <headword>[<constraint list>]

A headword of a UW is an English expression, a word, a compound word, a phase or a sentence of English. If the meaning of a headword is unique, the headword itself becomes a UW. Otherwise, constraints are attached to the headword to make more specific UWs. If a UW consists of a headword only, it is called a "Basic UW". For example, we can made different UWs

from the word “state”. The first one is **state(icl>abstract thing) to denote a kind of condition** that persons or things are in and **state(icl>country) denotes a country. We can also have state(icl>fix(agt>thing,obj>thing))** to denotes an action to fix the details of something and **state(icl>say(agt>thing,obj>thing)) to** denote. an action to say something. Here **icl, agt** and **obj** are UNL relations.

UNL relations

There are 46 relations in UNL such as **icl, gol, agt** and **obj**. they are used to connect every two UWs to construct the semantic networks of UNL expressions. The relations are edges in UNL graph and denote a semantic role of a UW or scope for others. The UNL binary relation has the following format.

rel (arg1, arg2)

rel represents the label of the UNL relation and its arguments arg1 and arg2 are Universal Words. The first argument acts as the parent of the relation whereas the second argument is the child of the relation. For example, in the UNL relation **agt (eat(icl>do).@entry, I(icl>person).@def))** , the label of the relation is **agt** and the arguments are **eat** and **I** where **eat** is parent of the relation and **I** is child of the relation. The description of the 46 UNL relations is given in annex A.

UNL attributes

Attributes are mainly for the purpose of describing subjectivity information. It includes time, aspect, emphasis, focus, topic, attitude, feeling and judgment. They are also used to specify qualities of concepts such as the generality, the specificity and the logicality of UWs. Attributes are attached to the UW or a scope to specify the semantic information. For example, to represent the object in the sentence ‘he bought books’; we have to refer the plurality of books bought using the attribute @pl. There are 87 attributes used to express the semantic of a sentence in UNL and are divided in to eight groups. These attributes are given in annex B.

UNL sentences

The UNL sentence is consisted of nodes or UWs interlinked with binary relations and modified by attributes. In UNL, s sentence is the smallest unit to be represented which means for each sentence, there is a corresponding UNL sentence or graph. There are two types of representing UNL sentences i.e. the table format and list format. But, the commonly used representation is the table format [36]. Table 2.5 and 2.6 shows the syntax of table format and list format respectively.

Table 2.6 syntax of UNL sentence in table format

<UNL sentence>	::=	<list of relations>

<list of relations>	::=	<binary relation>[<binary relation>...]
<binary relation>	::=	<relation>[“:”<Scope-ID>] “(” <source node>, <target node>“)”
<source node>	::=	<UW+attributes>
<target node>	::=	<UW+attributes>
<UW+attributes>	::=	<UW>{“:”<Scope-ID}[<attribute list>] “:” <UW-ID>

Table 2.7 Syntax of UNL sentence in list format

<UNL sentence>	::=	“[W]” <list of UWs> “[/W]” [“[R]” <list of relations> “[/R]”]
<UW+attributes>	::=	<UW+attributes> [<UW+attributes>...]
<UW+attributes>	::=	<UW>{“:”<Scope-ID>}[<attribute list>]“:”<UW-ID>
<list of relations>	::=	<binary relation>[<binary relation>...]
<binary relation>	::=	<source node><relation[“:”<Scope-ID>]<target node>
<source node>	::=	<UW-ID>
<target node>	::=	<UW-ID>

Where, ‘<’ and ‘>’ indicate a non-terminal symbol; ‘{’ and ‘}’ indicate a range; ‘[’ and ‘]’ indicate an omissible part; ‘...’ indicates more than 0 times repetition of the front part; ‘::=’ indicates that left part can be replaced by the right part and predefined delimiters are enclosed within “ ” symbol.

The table formats is illustrated in the following example sentence. The sentence is “I can hear the dog barking outside”.

{unl}
agt(hear(icl>perceive(agt>person,obj>thing)):06.@ability.@entry, I(icl>person):00.@topic)
obj(hear(icl>perceive(agt>person,obj>thing)):06.@ability.@entry, :01)
agt:01(bark(agt>dog):0H.@progress.@entry, dog(icl>mammal):0D.@indef)
plc:01(bark(agt>dog):0H.@progress.@entry, outside(icl>area):0P)
{/unl}

From the UNL representation of the sentence, ‘agt’, ‘obj’ and ‘plc’ are relations. ‘I(icl>person)’, ‘hear(icl>perceive(agt>person,obj>thing))’, ‘dog(icl>mammal)’, ‘bark(agt>dog)’ and ‘outside(icl>area)’ are UWs. ‘@ability’, ‘@entry’, ‘@indef’, ‘@progress’ and ‘@topic’ are attributes. The part “a dog barking outside” is expressed in a scope, ‘01’ is assigned as a Scope-

ID to the scope. Binary relations with the same Scope-ID appearing following the relation labels constitute the UNL Expression of a scope.

The UNL sentence can also be represented in a graph called UNL graph. The parent of a relation will be used as a root node in UNL graph. The UNL graph for the above sentence is given in fig 2.1.

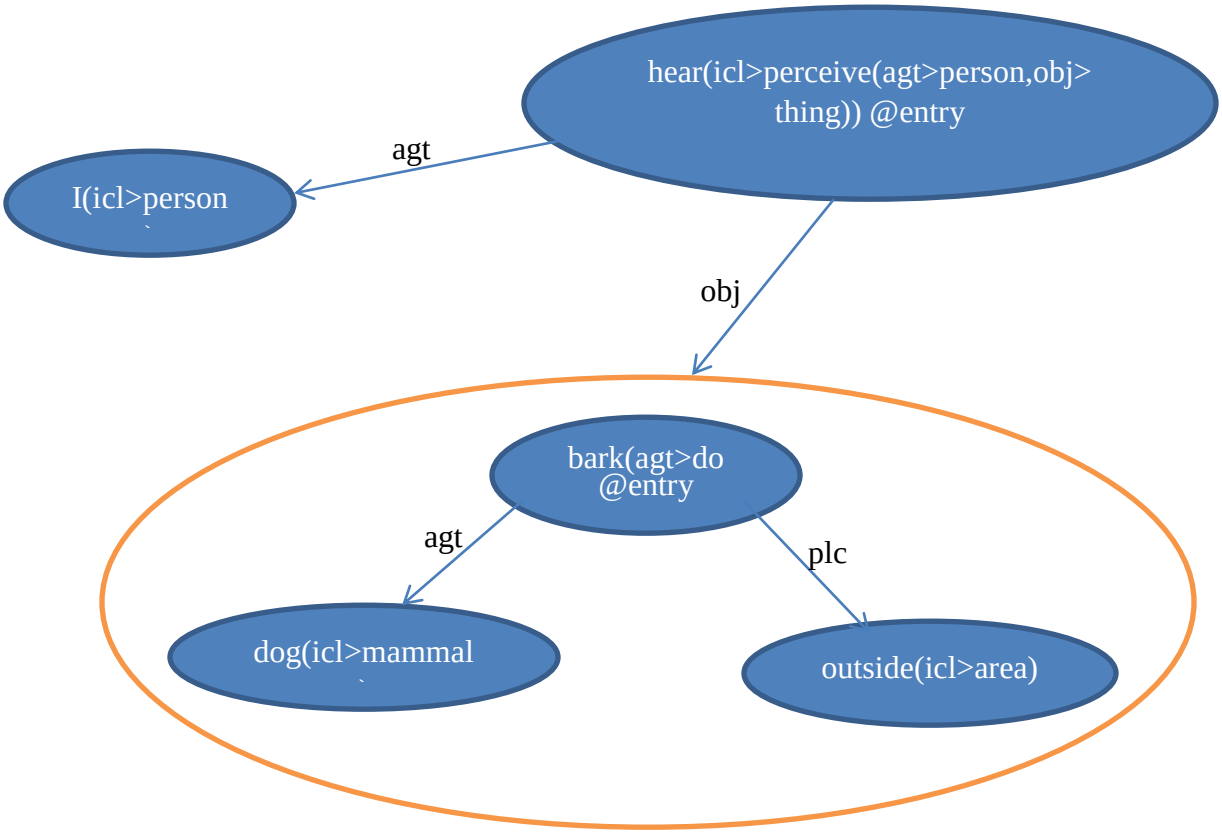


Fig 2.1 UNL graph

UNL EnConverter

The EnConverter is a software tool used to convert natural language sentences to UNL graph or UNL representation. The EnConverter works based on a word dictionary and a set of EnConversion rules (grammar rules of EnConversion). It analyzes sentences according to the EnConversion rules. It can deal with various natural languages by using respective word dictionaries and sets of EnConversion rules.

The EnConverter works in the following way. An input string of natural language sentence is scanned and all matched morphemes from the beginning of the string are retrieved from the word dictionary and become the candidate morphemes. Word selection is done by applying grammar

rules of EnConversion to these candidate morphemes. The rules will be applied to selected words to go through syntactic and semantic analysis in order to build a syntactic tree and a semantic network for the input sentence. This process is continued until it finishes all words of a sentence are inputted and analyzed. The output of this process is a semantic network expressed in the UNL format.

UNL DeConverter

The DeConverter is a software tool used to convert UNL graph or UNL representation of natural language sentences to equivalent natural language. It generates natural language sentences from UNL representation. It converts UNL expressions to various natural language sentences by using respective word dictionaries and a set of grammar rules of DeConversion of that particular language.

The DeConverter works in the following way. It first transforms the input of a UNL expression – a set of binary relations into a directed graph structure with hyper-nodes called node-net. The root node of a node-net is called entry node and represents the head (e.g. the main verb) of a sentence. DeConversion of a UNL Expression is carried out by applying DeConversion Rules to the nodes of node-net. It starts from the entry node, to find an appropriate word for each node and generate a word sequence (a list of words in grammatical order) of a target language. In this process, the syntactic structure is determined by applying syntactic rules, and morphemes are similarly generated by applying morphological rules. The DeConversion process ends when all words for all nodes are found and a word sequence of target sentence is completed.

2.8 Summary

Text summarization is a process of providing the central idea of an input document for different purposes. In this chapter, we have discussed about the current trends in text summarization systems. We have seen the types of text summarization systems and the stages involved in summarization systems. We have also discussed about the evaluation methods used for text summarization systems.

The most challenging part in text summarization system is the generation phase. To generate summary sentences that are not found in the input document involves the semantic representation of input sentences and analysis on the semantically represented sentences. One of the techniques used in semantic representation of natural language sentences is using UNL. In this chapter, we have seen how UNL is used to represent the semantics of natural language sentences.

Chapter Three: Related work

3.1 Introduction

This chapter discusses about the works done to automatic text summarization. We review research works that follow extractive and abstractive approaches for text summarization. We also present text summarization systems developed for Amharic language. After we discuss related works, we identify the gaps that we are going to bridge in our research work.

3.2 Extractive Text Summarization Systems

One of the first works done on text summarization is that of Luhn [3]. Luhn proposed that the frequency of a particular word in a text provides a useful measure of its significance. As a first step, words were stemmed to their root forms. Stemming is a process of changing words to their root forms or stems as different words derived from a single stem are assumed to have the same meaning. Stop words were also removed since they didn't indicate the relevance to the text. Stop words are words like conjunctions, pronouns or words that appear in many sentences, and thus do not serve to topically distinguish one sentence from another.

After stop words are removed and words are converted to their root forms or stems, the content words were compiled and sorted by decreasing frequency. On a sentence level, a significance factor was derived that reflects the number of occurrences of significant words within a sentence. All sentences are ranked in order of their significance factor, and the top ranking sentences are finally selected to form the summary.

Edmundson [37] proposed a domain specific single document summarization method that uses the techniques used by Luhn with additional features that rectify the process of selecting the most relevant sentences in the document. In addition to word frequency it uses cue phrases, sentence location and title and heading words to select sentences to be included in the summary.

Cue phrases are phrases that indicate the relevance of sentence to the topic or its irrelevance. Some phrases like 'significantly' and 'in conclusion' indicates the relevance whereas words like 'hardly' and 'impossible' indicates the irrelevance. Sentence location is another factor that indicates the relevance of sentence to the topic. For example, in news articles, the most important sentence is usually the first sentence whereas in technical documents, the most important sentences can be found in the conclusion section.

Title and heading words also have a significant contribution in selecting the most important sentences that are going to be included in the summary. Sentences that contain words from the title or heading of the document are important as authors usually use informative summaries that indicate the main idea of the whole document. By combining the features i.e. word frequency, cue words, title and heading words and sentence location, the system gives the extractive summary of a document.

Using lexical chain is the other method of summary extraction. It tries to solve the problem of anaphoric resolution in summarization systems that follow surface level approaches. Anaphoric expression refers to some parts of the text need to know their antecedents in order to be well understood. If a sentence containing an anaphoric link is extracted without the previous context, the summary can become difficult to understand.

The method introduced in [38] uses the WordNet thesaurus for determining cohesive relations between terms like repetition, synonymy, antonymy, hypernymy and holonymy that used to compose the chains of related terms. The score of a sentence is calculated based on the number and type of relations in the chain. Sentences where the strongest chains are highly concentrated are selected for the summary.

A similar method to the lexical chain where sentences are given a score according to objects or words they contain or mention is using co-reference resolution system [38]. Co-reference resolution is the process of determining whether two expressions in natural language refer to the same entity or not. The sentences where the occurrence of frequently mentioned objects exceeds the given limit are included in the summary.

MEAD [39] is a text summarization system developed at the University of Michigan that can generate single and multi-document extractive summaries. Its basic idea is to use three features that are centroid-based feature, position and overlap with first sentence or topic. The linear combination of the three features determines which sentences are most salient to be included in the summary. The aim is to similar sentences and the remaining sentences which are not similar to others will be included in the summary.

MEAD first uses topic detection and tracking system to identify all articles related to an emerging event to produce a set of clusters. The centroid will be built from each cluster and for each sentence, the values for the above three features will be computed. The centroid score measures how the sentences are close to the centroid. The position score measures how far the sentences are with respect to the beginning of a document. The overlap with the first sentence or topic is measured by calculating the tf*idf between the given sentence and the first sentence or the topic.

After obtaining the measure of the three features, all the scores are normalized and sentences which are too similar will be identified. The similarity of sentences is measured using cosine similarity measure on the scores of each sentence. Sentences that are similar to other sentences will be discarded. The remaining sentences that are not discarded will be included in the summary.

WebInEssence [47] is an example of summarization systems that have uses other than summarizing texts. It is a search engine to summarize clusters of related web pages which provide more contextual and summary information to help users explore retrieval results more efficiently.

It is a web based summarizer that uses a version of MEAD as its component. So, the features used to produce extracts are the same as the ones used in MEAD. The overall architecture of the system can be decomposed in to two components. The first component acts like a web-spider that collects URLs from the internet and it groups the URLs in to clusters. The second component performs the main stage that creates a multi-document summary from each cluster using the MEAD centroid algorithm.

NeATS [40] is a domain dependent text summarization system developed by information sciences institute of the University of Southern California. Its domain is the genre of newspaper news. The architecture of the system is composed of three components. These are content selection, content filtering and content presentation.

The main aim of content selection is to identify important concepts in the document. The techniques used in this stage are term frequency, topic signature or term clustering. The second component which is sentence filtering tries to filter sentences using different filtering mechanisms. Some of the techniques used are sentence position, stigma words and redundancy filter. NeATS uses a simplified version of MMR algorithm [40] for content presentation. Using MMR algorithm ensures the avoidance of redundancy. To ensure coherence of the summary, it outputs the final sentences in their chronological order.

NetSum [41] is a text summarization system that produces fully automated single-document extracts of news articles based on neuronal sets. It uses machine learning technique in which a train set is labeled so that the labels identify the best sentences. Then, a set of features is extracted from each sentence in the train and test sets, and the train set is used to train the system. The system is evaluated on the test set. The system learns the distribution of features of the best sentences from a train set and outputs a ranked list of sentences for each document.

3.3 Abstractive Text Summarization Systems

Automatic text summarization is a research area which is extensively covered but doing it in an abstractive technique is not well addressed. However, there are few research works that follow abstractive approach. Abstractive approach is an approach which is going to generate novel sentences that may not be present in the original text. The meaning of novel sentences may be a collection of the core meanings from two or more different sentences of the original text. In this section we will review some of the works done on automatic text summarization that follow abstractive approach.

GLEANS [42] is a text summarization system which is domain dependent that employ four different novel techniques to summarize document collections. The system first maps all documents in a collection into a canonical, database-like representation that makes explicit the main entities and relations in a document collection. The system also classifies each document collection into one of different categories. The categories may be depending on the domains in which the system is applied. For each type of document collection category, GLEANS generates

a short headline summary using a set of predefined templates. The headline summary constitutes the first two sentences in the abstract. The rest of the summary will be generated by extracting sentences that conform to a predefined schema from the database and presenting those sentences in an order that reflect coherent constraints specific to each category of document collection.

The system has different components that are integrated each other to form the overall system architecture. One of the components is database constructor. In this component, Contex which is a decision-based syntactic parser is used to parse all sentences in all of the document collections. Contex served as a named entity tagger. Then an extractor is used to map each parse tree into a canonical representation which makes explicit the surface string that corresponds to the sentence, the logical subject, the head verb and main complement of the sentence. The second component of the system is collection classifier which is used to produce summaries that are customized to the type of document collection given as input. The classifier applies hand-written rules that were developed by examining the collections in the training corpus. The rules use count measures that are collected over the most persons, places and events that are described in an input document collection, as tagged by Contex. Another component of the system is core entity/relation constructor. This component generates a small entity/relation database from the document collection type given by the classifier component. The small core/entity relation database contains the most salient words in the document collection.

The other component is headline generator. This component produces a customized headline by using the label assigned by the classifier to a given document collection and the core/entity relation database. For example, if we have a document collection called "Person" which is a collection of documents that are describing some person, the headline generator generates a headline using the template "The Story of *MainPerson*" where the *MainPerson* variable is replaced with the information extracted from the core entity/relation database. When the system is used for multi-document summarization, this component will be replaced by Lead sentences generator for multi-event collections which has a similar function with it. The main component of the system is abstract generator which is done by producing a library of schemas for abstract generation by analyzing all the abstracts in the training corpus manually. Each line in a schema corresponds to a single sentence in a summary. The last component is a post-processor component. This component removes the dangling discourse markers from the generated abstract, decide when to use pronouns and when to use full named entities, infers specific dates from relative temporal expressions and document time stamps and represents them in a canonical fashion.

Jing [23] developed a system for automatic sentence compression, or reduction that uses multiple sources of knowledge to decide which phrases in an extracted sentence can be removed, including syntactic knowledge, contextual information, and statistics computed from a corpus of professionally written summaries that are abstracts. Compression can be applied at any granularity, including a word, a prepositional phrase, a gerund, a to-infinitive or a clause. In the first step of reduction, the candidate sentences are parsed and all nodes in the tree that are

necessary in order to preserve the grammaticality of the sentence, such as the main verb or head of a noun phrase, are marked. Such nodes cannot be removed by the reduction module. Obligatory arguments of verbs are also marked. Contextual information is used in order to decide which parts of a sentence are most closely linked to the overall topic of the article and these parts will not be deleted even if it is syntactically possible to do so. The context weight for each word is computed as the number of links to the rest of the article in terms of repetitions, occurrence of morphological variants or of semantically related words identified using the WordNet database. Finally the likelihood of a human abstractor deleting a particular type of constituent is looked up in a table of pre-computed corpus statistics. These three factors are weighted and a constituent below a certain weight will be removed from the sentence. This approach is proved to be effective as evaluation demonstrated that 81% of the sentence reductions proposed by the system agreed with the reduction decisions made by professional human abstractors.

Galanis [49] proposed a system that automatically generates natural language summaries. The system propose two novel sentence compression methods which rewrites a source sentence in a shorter form, retaining the most important information of the original sentence. The first method is used to produce extractive compressions that are achieved by only deleting words. The second method is used to produce abstractive compressions that also involve paraphrasing. The experiment result shows that the extractive method gives comparable or better, in terms of grammaticality and meaning preservation, to those produced by other text summarization systems and the abstractive method of compression produces more varied (resulted by paraphrasing) and slightly shorter compressions than the extractive ones.

Extractive sentence compression method operates in two stages. In the first stage, multiple candidate compressions are produced by deleting branches from the dependency tree of the source sentence. To limit the number of candidate compressions, a trained Maximum Entropy classifier is employed that rejects unlikely actions i.e. unlikely branch deletions. In the second stage, an SVR model is used to select the best candidate compression, in terms of grammaticality and meaning preservation using mostly syntactic and semantic features.

Abstractive sentence compression does not just delete words and it operates in two stages. In the first stage, a large pool of candidate sentence compressions is generated. This pool consists of (a) extractive candidates, that are generated with the above extraction method and (b) abstractive candidates, which are generated by applying paraphrasing rules on the extractive candidates. In the second stage, the best candidates of the pool in terms of grammaticality are kept and they are ranked using an SVR model to select the best one. The feature set of this SVR includes language model scores, the confidence score of the extractive sentence compressor, the number of paraphrasing rules that have been applied, as well as features from word co-occurrence measures and Latent Dirichlet Allocation Models. For training and evaluation of different possible configurations of SVR, the author constructed a dataset that contains extractive and abstractive

candidates annotated with grammaticality and meaning preservation scores provided by human judges.

The system also present competitive sentence extraction method that assigns relevance scores to the sentences of the text that are going to be summarized which is coupled with a simple method to avoid selecting redundant sentences. The method assigns relevance (salience) scores to the input sentences using a support vector regression (SVR) model. The extraction method uses an SVR trained on examples whose target or ideal scores are calculated using n-gram similarity measures that are broadly used for summary evaluation.

Jing [24] proposed a summarization system using cut and paste approach to address the text generation problem in domain-independent single-document summarization. The approach is based on the fact that professional abstractors often reuse the text in an original document to produce an abstract summary. Human abstractors edit the extracted sentences from the input document and such editing operations are called revision operations. The proposed system simulates two types of revision operations that are frequently used by human abstractors. These are sentence reduction and sentence combination. Sentence reduction removes inessential phrases from sentences and sentence combination merges the remaining sentences and phrases together to produce novel sentences.

The sentence reduction module is the system which is specified on [48] and relies on multiple sources of knowledge to decide when it is appropriate to delete a phrase from a sentence. The sentence combination module relies on a set of rules to decide how to combine sentences and phrases and when to combine them. The main goal of sentence reduction is at improving the conciseness of generated summaries. Conciseness of a summary reflects the extent to which few words in a summary express the idea of large amount of words or sentences in the input document. The aim of the other component, which is sentence combination, is at improving the coherence of generated summaries.

The system also includes a Hidden Markov Model based sentence decomposition program that analyzes a corpus of human-written summaries. This program tries to identify where the phrase of an abstract summary originates in the original document. The program is trained using the aligned corpus of articles and summaries that are generated by professional abstractors.

3.4 Text Summarization Systems for Amharic

Many researches can be found on the literature that deals with automatic text summarization but Amharic language is not addressed well for text summarization system. However there are few attempts to develop text summarization system for Amharic language. But, to the best knowledge of the researcher, almost all of the researches follow the extractive approach for text summarization. In this section, we will review some of the works done on Amharic language even if they are of extractive approach.

The first work on text summarization for Amharic is done by Kemal [43]. Kemal proposed an extractive summarization system based on surface level approaches to weigh sentences. It uses statistical features such as presence of title words, stop words, cue phrases and frequent words to compute the weight of each sentence present in the document. Then top scoring sentences are selected by the system to form a summary.

The system also has a learning phase. In the learning, the system tries to update some of the features used to weigh sentences. This learning phase enable the system to use dynamic features for improved performance other than using static features. It enables the user to update the list of cue phrases and stop words dynamically and the system uses this updated list of features for improved measurement of the weight of a sentence.

The system is also trained with four news articles with their corresponding manual summaries. This training enables the system to learn the appropriate contribution of each statistical feature employed towards the importance of a sentence. Based on the training, the system adjusts the contribution of each feature to the weight of a sentence. Its evaluation results show that the use of title words and key words are important features to select sentences to be included in the summary.

Another work on text summarization system for Amharic language is done by Teferi [44] that employ machine learning technique. It uses Naïve Bayes classification technique by calculating the probability of each sentence to be included in the summary. The system has two phases, the training and test phase. The Naïve Bayes classifier is trained to identify the probability of a sentence to be included in the summary based on four features. The four features used are the presence of title words, location of sentences in the document, presence of cue words and presence of highly frequent words.

The system uses 480 news articles with their corresponding manual summaries for training and testing. In the training phase, 460 articles with their corresponding manual summaries are used. The aim of the training phase is to discover a classification function that accepts sentences as an input and gives the probability of the input sentence to be included in the summary based on the above specified four features. In the test phase, 20 documents with their corresponding manual summaries that are not present in the training set were used. The performance of the classifier is measured classification success rate, precision and recall. The evaluation results show that the use of single feature gives a poor result whereas using a combination of multiple features produce a better result in classifying sentences to be included in the summary or not.

Helen [45] produced a single document summarizer for Amharic legal documents. The system applied the same technique with that of Kemal [43] but the domain is in Amharic legal judgment documents. The summary starts by segmenting documents in to five parts. These are introduction, reason, fact, judicial analysis and decision. Then each sentence in the document is given a weight based on features like cue phrases and sentence location. Sentences with the highest weight are selected based on a given compression rate from each segments. Finally, the

selected sentences from each segment will be merged together to be used as a summary for the whole document.

Abraham [46] proposed multi-document summarization system for Amharic news documents. The system applied pure statistical approaches to extract sentences from multiple documents that will be included in the summary. The inputs are multiple news documents and are first undergoes the preprocessing stage. The system uses four statistical features to compute the significance measure of a sentence. These four features are context-sensitive frequency based feature, number of title words in a sentence, position of sentence in the text and centroid score of a sentence.

The centroid of a cluster will be built and represented by a vector of terms associated with their corresponding TFIDF value. Context-sensitive frequency based feature of a sentence is the score based on the average of the summation of the probability distribution of terms in the sentence. Probability distribution of terms is given by dividing the frequency of terms in the event set or cluster by the total number of terms in that event set or cluster. The centroid score of a sentence in the event set or cluster will be computed based on the cosine similarity of the sentence and the cluster centroid. Then sentences are given score based on the above specified features or a combination of all of the features and the top ranking sentences are taken as a summary.

The author suggests a mechanism to avoid redundancy of sentences in the summary as the system works on multiple documents. The cosine similarity of sentences will be computed and sentences with cosine similarity greater than the threshold value will be considered as redundant. When two sentences are identified as redundant based on the threshold value, the one with smaller computed score will be given another relatively small value. The new smaller score given will be less than the value given to other sentences which are not considered as redundant and this mechanism will give higher probability for new sentences to be included in the summary than sentences which are considered as redundant.

If redundancy is removed after each sentence is given a normalized score based on five mechanisms, the top ranking sentences up to the desired size are extracted and will be given as a summary. The size of the summary is computed based on the number of words rather than the number of sentences. This is because the author believes that words are finer than sentences to get better approximation of the size of the summary. The five mechanisms applied are based on the four features individually and the fifth one is the composite method or a combination of the four features.

To evaluate the proposed system, the author prepared 60 news items collected from three Amharic news providers. These news items are grouped into 20 event sets or clusters in which each event set consists of three news texts about the same topic. The system summaries were evaluated objectively and subjectively. In the objective evaluation, precision and recall were used and the result indicated that summarization based on Context Sensitive Frequency Based feature alone and summarization based on occurrence of title words feature alone performed

better than the others. In the subjective evaluation, the redundancy and the linguistic quality of the system summary were assessed and the results were shown to be promising.

Another important system on the field of text summarization for Amharic language is done by Melesse [4]. It is domain dependent text summarization system that works on the domain of news texts. The system tries to solve the problem of text summarization systems that are based on pure statistical approaches and machine learning algorithms. Text summarization systems that follow pure statistical approaches fail to capture the main topic of the document because they ignore the semantic of words in the document whereas that of machine learning algorithms is very costly since it requires a great deal of training corpus. Another problem of machine learning algorithms for text summarization systems is the adaptability of the techniques for another domains or languages.

The system employs Latent Semantic Analysis (LSA) to overcome the above specified problems. LSA has the capability of finding the semantic relations among words and sentences of the input document. LSA can be applied for text summarization systems for finding out the concepts and finding representative sentences for those concepts. Those sentences that have greater relation with the important sentences are included in the summary. The overall assumption of using LSA is that the right singular matrix obtained after SVD decomposition of a term by sentence captures the salient topics of the document. Then, summarization can be done by selecting the most important sentences that reflect those identified topics. The author proposed two algorithms that are based on LSA. These are TopicLSA and LSAGraph.

TopicLSA is used along with document genre information to select semantically important sentences to be included in the summary. First, the term by sentence matrix will be constructed to be analyzed by singular value decomposition (SVD). The author includes the title of the document to the term by sentence matrix as terms in the title affect the construction of the latent semantic space. After constructing the term by sentence matrix U, it will be used to extract the important concepts or topics of the document. For each column of the matrix U* Σ, the top m terms that have high index value in the column are selected and all the resulting terms are concatenated to form a topic vector. After identifying the topic vector, it will be folded in to the latent space of U* Σ for similarity comparison of all of the sentences against the topic vector. Then, all sentences of the document will be given three scores.

The first score is the cosine similarity of the sentence with the topic vector in the latent space. It will be used as an indication of the importance of sentences as more similar sentences with the topic vector are more likely to be included in the summary. The second score is the cosine similarity of sentences with the title of the document in the latent space. This score describe the fact that headlines of news texts carry the general idea of the document as a whole. The third score describes the position of each sentence. It will be given as $1/n$, where n represent the distance of sentences from the beginning of the document. It is based on the fact that news texts put the most important sentences near the beginning of the document. It shows that the

importance of sentences is inversely proportional to the distance of sentences from the beginning of the document. The total result of the three scores will be given for each sentence and sentences that have the highest score will be included in the summary.

The second algorithm which is LSAGraph combines both LSA and Graph-based approaches to select the most important sentences to be included in the summary. Graph-based algorithms use graphs to represent a document in which each sentence is represented by vertices of a graph and the edges represent the similarity of sentences. Then sentences will be ranked based on their importance to the graph. The importance of sentences to the graph is determined by the similarity of the sentence to other sentences in the document. So, the similarity measure of a sentence has a greater impact on the summary result. Thus, the author proposed LSA to be used as a similarity measure to get a better result since LSA deals with the semantic meaning of concepts when measuring the similarity.

Habtamu [9] proposed multi document summarization system based on Probabilistic Latent Semantic Analysis (PLSA). PLSA is a variant of LSA that has a sound statistical foundation and define a proper generative model of the data. The author proposed different text summarization algorithms to create a summary by extracting sentences from multiple documents that are written on the same topic. The algorithms depend on sentence by topic matrix to select sentences.

The first stage of the system is preprocessing since it is required for better performance of text summarization systems. The preprocessing module is responsible for performing tasks such as lexical analysis, normalization, stemming, stop-word removal and term extraction. The second component of the system is topic identification process. This includes the construction of term by sentence matrix, construction of sentence by sentence matrix and construction of probabilistic latent semantic analysis (PLSA) models that are used to generate topic based features that are going to be used as a ranking mechanism for the relevance of sentences. The third component is sentence ranking component and is responsible for giving rank based on the relevance of sentences to be included in the summary depending on the features selected by topic identification process. This component has different sub-components. One of the sub-components is score averaging. This component is to calculate the average score of each sentence based on the algorithms employed. The other component is redundancy removal and is responsible for checking if two or more sentences have similar idea. When the redundancy removal module encounters two or more sentences with similar idea, it will discard all similar sentences except one of them. The final component of the system is sentence extraction module and it is responsible for extracting the top ranking sentences to be included in the summary.

3.5 Summary

We have reviewed different works on automatic text summarization systems that are related to our study. From the review, we observe that automatic text summarization systems that follow extractive approach are done using different algorithms and techniques. Automatic text summarization systems that are of extractive approach employ different techniques to select the most important sentences from the given document that are going to be included in the summary. There are varieties of techniques used by researchers like statistical methods, graph-based algorithms, machine learning approach and Latent semantic analysis etc. for representing input sentences and select the sentences to be included in the summary.

But, automatic text summarization systems that follow abstractive approach are not well addressed as much as systems that are of extractive approach. However, there are different attempts to develop automatic text summarization systems that are of abstractive approach for different languages. The difficulty of abstractive text summarization systems is the generation phase. The generation phase involves modeling the grammar and the morphology of the natural language involved.

Amharic is also a language which have many researches on automatic text summarization that follow extractive approach. But, doing the text summarization process in abstractive approach is not addressed so far. This is due to the complexity of the morphology of the language. Since Amharic is one the morphologically complex languages, it is difficult to model the morphology and the grammar that can be used in the summarization process.

Thus, in this thesis we attempt to develop an automatic Amharic text summarizer that follow abstractive approach. We use UNL expression to represent the semantics of Amharic sentences. The grammar and morphology of Amharic language is also considered in developing the summarizer since the generation phase involves modeling the grammar and the morphology of the language used.

Chapter Four: Design of Amharic Text Summarizer using Abstractive Approach

4.1 Introduction

This chapter describes about the proposed design for Amharic text summarizer using abstractive approach. The design for the summarizer has five major components which are text preprocessing module, sentence clustering module, Natural Language (NL) to Universal Networking Language (UNL) converting module, UNL analysis module, Universal Networking Language (UNL) to Natural Language (NL) converting module.

The interaction between components of the proposed architecture is also presented along with the structure of different databases on each phase. Each component in the proposed system is also described using an example.

4.2 Proposed Architecture

As described in the beginning of this chapter, the summarizer developed in this thesis has five components which are preprocessing module, sentence clustering module, Natural Language (NL) to Universal Networking Language (UNL) converting module, UNL analysis module, Universal Networking Language (UNL) to Natural Language (NL) converting module. The general architecture of the summarizer is given in figure 4.1 and the detail description of each component is explained in the succeeding sections.

The text preprocessing consists of indexing, normalization, stop word removal and stemming. The indexing is a process of giving an index for each sentence and storing each sentence with their index in separate repository. The purpose of storing sentences in repository is for the process of EnConversion since the text preprocessing changes the originality of the sentence and we need the original sentences for representing the semantics of each sentence using UNL expression. The normalization, stop-word removal and stemming processes are used as preparation for clustering module. The sentence clustering is a process of grouping semantically similar sentences in one cluster. This process uses the Wordnet repository to determine the semantic relation of words.

The Natural Language (NL) to Universal Networking Language (UNL) conversion is a process of converting natural language sentences in each cluster to corresponding UNL representation of sentences. The UNL analysis is a process of getting a single UNL representation for each cluster since each cluster consists of semantically related sentences. It generates a single UNL representation for each cluster and passes it to the next component which is UNL to NL conversion. This component then generates the NL sentences from the UNL representation. Each cluster will have its own NL sentence which is the central idea of sentences in the cluster.

The general architecture of the proposed system is presented in figure 4.1 and the description of each of the components of the proposed system is also presented in subsequent sections with examples.

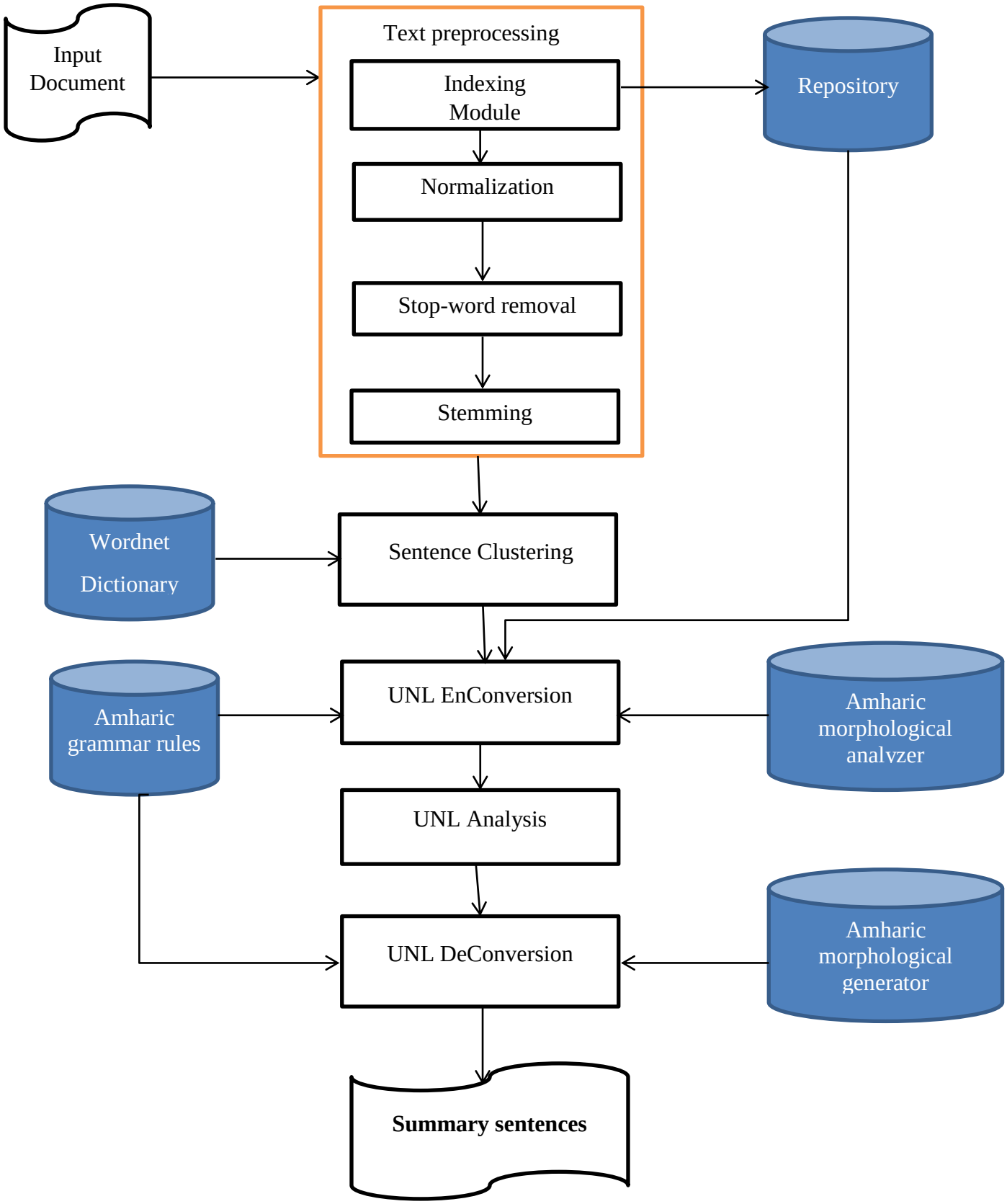


Fig 4.1 Architecture of Amharic text summarizer

4.3 Text Preprocessing

The preprocessing module is used to prepare the text in a format that is suitable for the steps involved in the summarization process. Using a preprocessing module in natural language applications has an impact on the performance of the system. It affects the performance in positive manner because the input will be formatted in such a way that it will be easier to be used in succeeding steps of the process.

In this thesis, the text preprocessing module consists of steps such as indexing, normalization, stop-word removal and stemming. Some of the components of the preprocessing module are adopted from the work of Tessema [14].

The indexing step is used to store the original sentences in a repository for later use in converting the Natural Language (NL) sentences to Universal Networking Language (UNL) since we need the semantics of the sentence to be represented. The normalization, stop-word removal and stemming steps are performed for the clustering module because similar sentences need to be in the same cluster.

Indexing

Indexing sentences is the first step of preprocessing the input document in this thesis. It is needed to separate each sentence of the input document and store in a repository for the use of the components of the system. In our design we need the original sentences later for semantic representation and before going through the processes that change sentences from their original form to other forms. In Amharic language, sentences are identified by the delimiters "።", "?" and "!". Words are also identified within a sentence by space and a number of punctuation marks such as "፡", "፤" and "!" [10].

Normalization

Normalization is the process of changing a form of a text to another form. In natural language some words may have different representation still conveying the same message. For efficient processing, words with different representation should be normalized to a single representation. Amharic language has two issues of normalization issues [14]. The first one is identification and replacement of Amharic words that have the same use and pronunciation but different representation. For example the words "ሀሜት" and "ሐሴት" have similar usage but different forms. The replacement is made by using a representative alphabet from a set of similar alphabets.

The second issue is identification and replacement of shorter forms of a word that is written using period (".") or forward slash ("/"). An example is the replacement of "ወ/ሮ" by "ወይዘሮ". The list of short words and there normalized form can be found from appendix A [14].

Stop-word Removal

Stop-words are words that do not add information but are necessary for gluing sentences. These include prepositions, articles, pronouns, conjunctions, etc. stop-word removal is done by using predefined set of words that should be removed from a sentence. Since stop-words are predefined, they are language dependent. Amharic language has its own stop-words such as "ሆነ", "ነው", "ነበር", etc. Tessema [14] provided a list of stop-words for Amharic language and can be found in appendix B.

Stemming

Stemming is a process of changing the original form to its root form (stem). In a text, words can have different forms such as plural or singular, present or past tense, which are derived from the same root. Amharic language is one of highly inflected languages and the use of stemmers is required to get the root of words. For example the various words "ሀገር", "ሀገራችን", "ሀገራቸው", "ሀገሬ", "ሀገሩ", "ሀገሯ", "ሀገርህ", "ሀገራችን" are changed to their stem word "ሀገር".

As it is explained in the beginning of the chapter, the stemming process is used to make the clustering step more effective. The clustering module uses a Wordnet for the relation of words in different sentences. Since words in the Wordnet are in their root forms, the words in the input sentence should be converted to their root forms.

4.4 Sentence Clustering

The need for clustering process is to have clusters of sentences which have similar and highly related meaning. Here the semantics of sentences should be put in to consideration because, unlike summarization systems which follow extractive approach, our model is abstractive approach and have to consider the meaning of sentences to put in a cluster that consists of similar or highly related sentences. The clusters are needed to generate a single representative sentence for each cluster. For clustering module, we use predefined Wordnet database and proposed a new algorithm.

The Wordnet dictionary is based on root words and it shows the stems that can be formed from the roots. In Amharic language, a root form of a verb may have different stems that form infinite words from them. The relation of roots and their derived stems are stored in the database so that when using the Wordnet dictionary, we can use either the root form of a word or the stem of a word. Table 4.1 shows the structure of the Wordnet dictionary which shows the stem of a root word and its relation with other roots.

Table 4.1 Structure of a Wordnet dictionary

Root word	Stems	Synonyms
ንድፍ	ነደፍ, ነዳፍ, ንደፍ, ነዳደፍ, ነድፍ	ቅርፅ

ድግፍ	ደገፍ, ደጋፍ, ደጋገፍ	እግዘ, ‘ርድ
ምት	መት, መታት, ምታት	ድብድብ
ፍጽም	ፈጸም, ፈጻጸም, ፈጻም	ድርግ

The stem and the word for Amharic nouns is usually the same. But, verbal roots in Amharic has language has many stems and from this different stems an infinite number of words can be generated. The morphological complexity of the language can be seen from the derivation of these words from a single root word.

The clustering algorithm is designed in such a way that it determines the number of clusters and sentences in the cluster based on the input text and by using the Wordnet database. It is done by constructing two matrixes. The first one is a term-by-sentence matrix in which the column represents sentences and the row represents terms in the input document and the matrix values are the relation of terms to the sentence.

Table 4.2 term-by-sentence matrix

	Sent 1	**Sent 2**	**Sent 3**	.	.	.	**Sent n**
Term 1	V11	V12	V13	.	.	.	V1n
Term 2	V21	V22	V23	.	.	.	V2n
Term 3	V31	V32	V33	.	.	.	V3n
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.
Term n	**Vn1**	**Vn2**	.	.	.	.	Vnn

The second matrix is a sentence-by-sentence matrix in which both the column and row represent sentences and the values of the matrix are similarities of sentences. The terms are in their root form since the input document will go through preprocessing step before the clustering. This will be easier for our algorithm to use the Wordnet database which contains root words of Amharic

languages and their relation with other words of Amharic language. The procedure for clustering is given in table 4.3.

Table 4.3 Algorithm for clustering of sentences

- ***Accept input sentences with their index and Wordnet dictionary***
- ***Construct term by sentence and sentence by sentence matrix***
- ***Fill the values of term by sentence matrix***
  - ***If a term occurs in a sentence, give the value 1*no of occurrence***
  - ***If a term's synonym occurs in a sentence, give the value 1*no of occurrences***
  - ***Sum the two values***
- ***Fill the values of sentence by sentence matrix with values from 0 to 1***
  - ***If all terms of a sentence found in the sentence, give the value 1***
  - ***If none of the terms of a sentence and their synonyms found in the sentence, give the value 0***
  - ***Depending on the number of terms and their synonyms occurrence give a value between 0 and 1.***
- ***Group sentences with higher value of relation using the two matrix values in the same cluster***
- ***Return sentence indexes with their cluster number***

As we can see from the clustering algorithm in table 4.3, step 5 involves many iterative processes depending on the number of sentences in the given input document. It starts from the first sentence and puts it in one cluster. Sentences which have meaning relation will be added to the cluster based on the value of the two matrixes. When encountered a sentence with no relation with the first sentence, it will be added to another cluster and other sentences will be added depending on the relation value with the sentences in the cluster. Here, a sentence may be moved to other clusters after it is added in to one cluster. This is because of the use of Wordnet to determine the meaning relation of words and one sentence may have a relation to sentences in one cluster and a higher relation to sentences in other cluster. In this case, the sentence will be moved to the cluster in which it has sentences with higher value of relation to the given sentence.

4.5 UNL EnConversion

The UNL EnConverter is a software tool used to convert natural language sentences to UNL representation. It uses the grammar rules of a language, word dictionary and EnConversion rules. Originally, the EnConverter is used in machine translation to be used in representing the natural language sentences of a source language in to equivalent UNL representation.

Since the EnConverter is originally designed for machine translation, it has different components like UNL ontology, word dictionary and EnConversion rules that are used to represent the meaning of the given natural language sentence. The UNL EnConverter in our proposed system uses the grammar rules of Amharic language and morphological analyzer for Amharic words to determine different parts of the UNL representation. The grammar rules are used to determine the universal words and relations whereas the morphological analyzer shows the attributes to universal words and relations.

The EnConverter component converts one natural language sentence to UNL expression at a time. Each word in a sentence is given to the morphological analyzer in order to get all the morphemes and the stem. We have used the morphological analyzer developed by Gasser [50] with a little modification. The modification is needed because the morphological analyzer is not 100% correct. We have used a learning approach where new words are encountered; their morphological information of the word is stored for future use. The structure of a morphological database is shown in table 4.4.

Table 4.4 structure of morphological database

Word	Stem	Morphemes
አልመጣችሁም	መጣ	አል -, - አችሁ, - ም
ሀገራት	ሀገር	- አት
የልጆቻቸው	ልጅ	የ -, - አች, - አቸው
ሰበረላት	ሰበር	- ኧ, - ል, - አት
አልፈለጋትም	ፈለግ	አል -, - አት, - ም

As it is shown from table 4.4, each word is represented in its stem and different morphemes that are used to inflect the word for different attributes. The morphemes are used for choosing from different attributes that are used in UNL expression. Different markers of a word that are indicated by the morphemes are also used to choose from the relations exist in UNL expression. The mapping between the morphemes of a word and the UNL relations and attributes has rules that are stored in database. Table 4.5 shows some examples of mapping from different morphemes into UNL attributes. Morphemes may change their properties depending on whether they are applied to a noun or a verb.

Table 4.5 properties of different morphemes

Morphemes	Applied in to	Meaning/attribute
አል -	Verb	Negativity
- ኣት	Verb	The object of the verb is feminine
- ኣት	Noun	The noun is in plural form
- ኣችሁ	Verb	Plural or politeness
- ኧ	Verb	Masculine

Some morphemes have different meaning when they are applied to different lexical meanings. For example, the morpheme - ኣት can be used as a suffix for both nouns and verbs. The noun ሀገራት has the suffix - ኣት and the verb ወሰደላት also has the suffix - ኣት. The suffix - ኣት in the noun ሀገራት indicates that the noun is plural whereas in the verb ወሰደላት indicates that the object is a feminine noun.

We have designed an algorithm for the process of EnConversion. The algorithm uses a natural language sentence as an input and outputs the UNL expression equivalent to the meaning of the natural language inputted. The algorithm is given in table 4.6.

Table 4.6 Algorithm for EnConversion rules

1. ***Accept input Natural Language sentence***
2. ***Determine the main verb of a sentence using the grammar rules***
3. ***Find the relation of other words with the main verb***
   - ***If another verb is found***
     Find the relation of other words to this verb
   - ***If not***
     Go to step 4
4. ***Determine the attributes of each word using the morphological analyzer***
5. ***Construct a relation between words with their attributes using UNL rules***
6. ***Return the UNL expression***

As it can be seen from the algorithm, the EnConverter module uses the grammar rules of the language and the morphological analyzer to determine the universal words and their relation together with their attribute.

For example, let's consider the sentence አበበ ትናንት መጽሃፎቹን ሸጣቸው and see the steps of the algorithm. Using the grammar rules of Amharic language, which says Amharic sentences are in

SOV (Subject-Verb-Object) order, we can determine that ሸጣቸው is the main verb of the given sentence. The next step is to find the relation of other words of a sentence to the main verb. አበበ is a noun which is the subject of the sentence and መጽሃፎቹን is a noun and the object of the sentence. ትናንት is an adverb which indicates the time of the action.

After determining the relation of words with the main verb, we use the morphological analyzer to get the attributes of each word. አበበ is a noun which is a male person and ትናንት is an adverb which indicates the time of the action. These two words are found simply since we have a list of male persons and a list of an adverb. When we analyze the word መጽሃፎቹን, we have the main word መጽሃፍ and morphemes - ኦች, - ኡ and - ን used as a suffix to the main word. The main word መጽሃፍ is a noun. The suffix - ኦች indicates that the noun is plural; - ኡ indicates that the noun is definite and the suffix - ን indicates that the noun is inflected for genitive case. The word has as the main word ሸጥ and the morphemes - ኧ and - ኣቸው are used as a suffix to the main word. The main word ሸጥ is a verb which has the meaning 'sell'. The suffix - ኧ indicates that the subject of the verb is male and the suffix - ኣቸው indicates that the object of the verb is plural.

After determining the attributes of each word, we will construct the relation using the EnConversion rules. Here, the main verb is ሸጥ and has the meaning 'sell'. We can find from the EnConversion rules; the word 'sell' is a kind of action (do) which has an agent which is the doer of the actin and an object which is the receiver of the action. Here, the noun አበበ will be the agent and መጽሃፍ will be an object whereas the adverb ትናንት indicates the time of the action.

The UNL expression will be derived from the above information and using the UNL rules of representing universal words or concepts, relations and attributes. The resulting UNL expression from the give sentence is given in table 4.7.

Table 4.7 UNL expression for the example sentence

{unl}

agt('ሸጥ' sell(icl>do(agt>person, obj>thing).@plu.@entry, አበበ(icl>male person))

obj('ሸጥ' sell(icl>do(agt>person, obj>thing).@plu.@entry, መጽሃፍ(icl>thing).@plu.@def)

tim('ሸጥ' sell(icl>do(agt>person, obj>thing).@plu.@entry, ትናንት(icl>past time))

{/unl}

The UNL graph for the above example is shown in figure 4.2 and is derived from the UNL expression. The UNL graph is a tree in which the root node is the main verb of the sentence and all words of the sentence correspond to a particular node in the UNL graph.

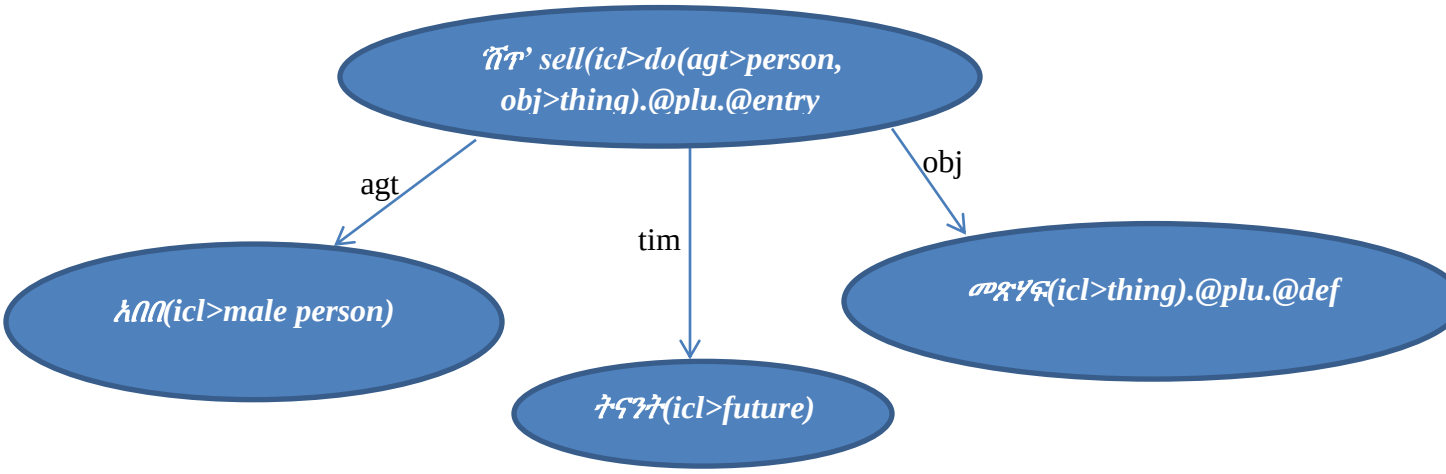


Fig 4.2 UNL graph for the UNL expression in table 4.4

We can see an example of a group of sentences to see the process of EnConversion to make it clear that how the component processes of the proposed architecture. As we can see from the architecture of the proposed system in figure 4.1, the input for the EnConversion process is a group of natural language sentences that have some sort of similarity. We consider the following five sentences in a single cluster as an input for the EnConversion process.

አልማዝ መጽሃፍ ማንበብ ትወዳለች::

አልማዝ መጽሃፍ የምትገዛው ከአበበ ሱቅ ነው::

አልማዝ እስካሁን ብዙ መጽሃፍ ገዝታለች::

የነአልማዝ ቤት በመቃጠሉ ብዙ ንብረት ወድሟል::

ከተቃጠለው ንብረት መካከል ብዙ መጽሃፍቶች ይገኙበታል::

The UNL EnConversion process takes each of the input sentences separately and converts each of the sentences to respective UNL expression. First, the sentence አልማዝ መጽሃፍ ማንበብ ትወዳለች will be analyzed to get the main verb of the sentence. Since Amharic has a subject-object-verb (SOV) arrangement, the verb of the sentence is ትወዳለች and the subject is አልማዝ whereas the object is መጽሃፍ ማንበብ which is a compound word. The compound word is composed of the words መጽሃፍ which is a noun and ማንበብ which is a verb. This shows the relation between words of the sentence with the main verb.

Then, we use the morphological analyzer to determine the attributes of each universal word in UNL expression. The morphemes of the main verb ትወዳለች are the prefix ት - to indicate the verb is in declarative mode and it is inflected for feminine noun. The suffix - ች indicates that the action of the verb is continuous and it is also inflected for feminine noun. The prefix - መ added in the word ማንበብ indicates that the verb is infinitive. But the nouns አልማዝ and መጽሃፍ do not have any morpheme.

So, the resulting UNL expression for the first sentence which is አልማዝ መጽሃፍ ማንበብ ትወዳለች will become

{unl}

agt('ውድድ'(icl>do(agt>person, obj>thing).@ability.@entry, አልማዝ(icl>female person).@topic)

obj('ውድድ'(icl>do(agt>person, obj>thing).@ability.@entry, :01)

obj:01('ንባብ'(icl>thing).@progress.@entry, መጽሃፍ(icl>thing).@indef)

{/unl}

The UNL EnConversion process for the rest four sentences is done like it is done for the first sentence. The resulting UNL expression is presented along with the natural language sentences.

አልማዝ መጽሃፍ የምትገዛው ከአበበ ሱቅ ነው

{unl}

agt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, አልማዝ(icl>female person).@topic)

obj('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, መጽሃፍ(icl>thing).@indef)

plc('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, :01)

pos:01('ሱቅ'(icl>place).@indef.@entry, አበበ(icl>male person).@topic)

{/unl}

አልማዝ እስካሁን ብዙ መጽሃፍ ገዝታለች

{unl}

agt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, አልማዝ(icl>female person).@topic)

obj('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, መጽሃፍ(icl>thing).@indef.@plu)

tmt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, now(iof>time).@present)

{/unl}

የነአልማዝ ቤት በመቃጠሉ ብዙ ንብረት ወድሟል

{unl}

obj('ወደመ'(icl>thing).@past.@entry, ንብረት(icl>thing).@plu)

rsn('ወደመ'(obj>thing).@past.@entry, :01)

obj:01('ቃጠሎ'(obj>thing).@past.@entry, :02)

pos:02('ቤት'(obj>thing).@entry, እነ አልማዝ(iof>persons).@indef)

{/unl}

ከተቃጠለው ንብረት መካከል ብዙ መጽሃፍቶች ይገኙበታል

{unl}

obj('ቃጠሎ'(obj>thing).@entry, :01)

pof:01('መጽሃፍ'(icl>thing).@def.@plu, ንብረት(icl>thing).@def)

{/unl}

Finally, the UNL expressions of sentences that are in the same cluster will be passed to the next component which is the UNL analysis.

4.6 UNL Analysis

This is the most important component in our proposed system and it accepts UNL expressions of many sentences and tries to analyze the expressions and produce a single expression that could be used as the main idea of each UNL representation. This is done by using the properties of the components of the UNL expression. The components of the UNL expression are the universal words, the relations of universal words and attributes of the universal words. From these components, one can easily found the main idea of each sentence.

The task of the UNL analyzer component will be easy since the clustering module gives the most related sentences in meaning within a single group. So, if sentences with the same or related meaning are within the same cluster, the algorithm is used to pick representative UNL expression from a group of UNL expressions.

The UNL analyzer works in the following way. It first accepts different UNL expressions and based on the components of the UNL which are the universal words, their attributes and the relation between universal words it tries to find a common meaning between the UNL expressions.

The algorithm for the UNL analysis process tries to find the most repetitive universal words from the group of UNL expressions. Then the relation of those repeated universal words will be analyzed from different UNL expressions to get their relation. The relation between other universal words and those repeated universal words should also be analyzed to get new relations that can represent many relations. The other thing we have to consider is the transitive properties of some of the relations of UNL expressions. This will lead to a UNL expression for generating new sentences that are not present in the input sentences.

For example, we can consider a group of sentences we used as an example for the EnConversion process. The input for the UNL analysis is the UNL expressions of a group of sentences. So, the input will be five different UNL expressions. These are

{unl}

agt('ውድድ'(icl>do(agt>person, obj>thing).@ability.@entry, አልማዝ(icl>female person).@topic)

obj('ውድድ'(icl>do(agt>person, obj>thing).@ability.@entry, :01)

obj:01('ንባብ'(icl>thing).@progress.@entry, መጽሃፍ(icl>thing).@indef)

{/unl}

{unl}

agt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, አልማዝ(icl>female person).@topic)

obj('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, መጽሃፍ(icl>thing).@indef)

plc('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, :01)

pos:01('ሱቅ'(icl>place).@indef.@entry, አበበ(icl>male person).@topic)

{/unl}

{unl}

agt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, አልማዝ(icl>female person).@topic)

obj('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, መጽሃፍ(icl>thing).@indef.@plu)

tmt('ገዛ'(icl>do(agt>person, obj>thing).@past.@entry, now(iof>time).@present)

{/unl}

{unl}

obj('ወደመ'(icl>thing).@past.@entry, ንብረት(icl>thing).@plu)

rsn('ወደመ'(obj>thing).@past.@entry, :01)

obj:01('ቃጠሎ'(obj>thing).@past.@entry, :02)

pos:02('ቤት'(obj>thing).@entry, እነ አልማዝ(iof>persons).@indef)

{/unl}

{unl}

obj('ቃጠሎ'(obj>thing).@entry, :01)

pof:01('መጽሃፍ'(icl>thing).@def.@plu, ንብረት(icl>thing).@def)

{/unl}

From the input UNL expressions, we can see that the most repeated universal words are አልማዝ and መጽሃፍ with different properties and relations in different UNL expressions. From different relations of these universal words, we have to determine the relation between these two universal words. There are many relations that connect these two universal words አልማዝ and መጽሃፍ like in ***'ገዘ'(icl>do(agt>person, obj>thing).@past.@entry*** as an agent and object respectively. From these different relations we can see that the agent አልማዝ is the owner or possessor of the object መጽሃፍ and we proceed to the relation of other universal words with these two.

Here, we can find transitive relations between the universal words መጽሃፍ and ቃጠሎ in the relations ***obj('ቃጠሎ'(obj>thing).@entry,:01) and pof:01('መጽሃፍ'(icl>thing).@def.@plu, ንብረት(icl>thing).@def)*** that shows the universal word መጽሃፍ is part of the universal word ንብረት. The other relation is between the universal word ንብረት and ወደመ which indicates that the universal word ንብረት which is a noun has a process of ወደመ because the universal word ወደመ is a verb. From these relations, we can conclude that the noun መጽሃፍ has also a relation with the verb ወደመ and this will give us a new UNL expression.

The new UNL expression has three universal words which are ወደመ , መጽሃፍ and አልማዝ with the relation pos which indicate that something is possessed by someone and the relation of the verb ወደመ with the object መጽሃፍ.

Finally, the UNL expression will be built from the relations and their attributes. The attribute of the universal word ወደመ will be ***@past*** to indicate the action is past tense. the attribute of the universal word መጽሃፍ will be @plu and @def to indicate that there number which is plural and the definiteness respectively. The resulting UNL expression will be as follows

{unl}

obj('ወደመ'(obj>thing).@entry,@past, :01)

pos:01('መጽሃፍ'(icl>thing).@entry.@def.@plu, አልማዝ(icl>female person)

{/unl}

4.7 UNL DeConversion

The UNL DeConverter is a language independent generator that can convert UNL expressions into a variety of natural languages. The DeConverter uses word dictionaries and sets of grammar rules of DeConversion for the target language. A word dictionary contains the information of

words that correspond to universal words in the UNL expression and the grammatical attributes or features that shows the behaviors of the words.

The DeConverter component in our proposed system does the reverse process of what the EnConverter component do. It uses morphological generator of Amharic language and grammar rules of the language. It uses morphological generator to determine the appropriate word representation for each nodes of the UNL graph. The grammar rules are used to determine the correct sequence of words in a sentence.

The algorithm used for the DeConverter of our proposed system is given in table 4.8

Table4.8 Algorithm for DeConversion rules

1. ***Accept UNL expression of a sentence***
2. ***Generate Natural Language words from universal words of the UNL expression with their attributes using the morphological generator***
3. ***Construct Natural Language words or phrases from each relation of the UNL expression using the arguments and their attributes***
4. ***Use the grammar rules of the language to determine the correct position of each word or phrase***
5. ***Remove the redundant verbs from different phrases with appropriate morphemes***
6. ***Return the NL sentence***

As we can see from the algorithm, the components of the UNL expression which are universal words, their attributes and relations between universal words provide the information for the morphological generator. The morphological generator used the above information to determine different morphemes that are appropriate to the meaning of the sentence. The grammar rules of the language determine the position of each word in order to have the appropriate meaning. For example, let's consider the following UNL expression and examine the steps of the algorithm.

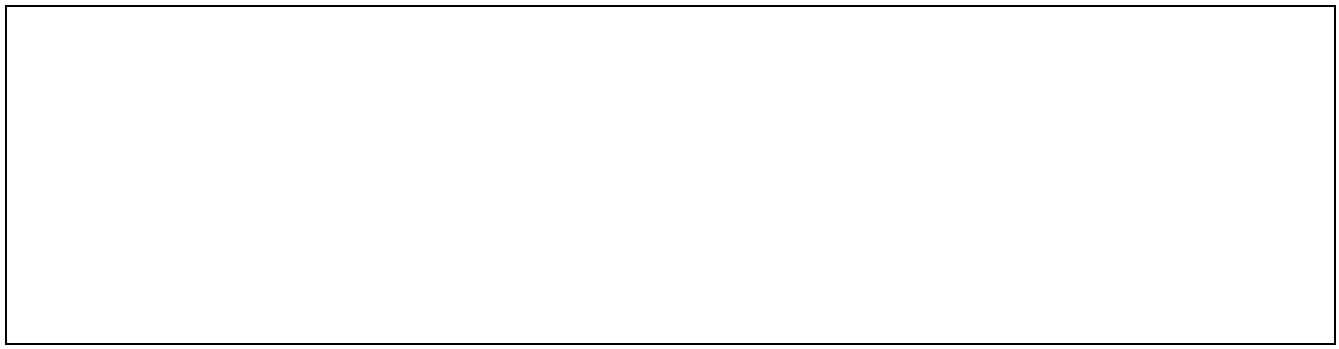
{unl}

agt('አባረረ' (icl>do(agt>person, obj>thing).@entry, አስቴር(icl>female person))

obj('አባረረ' (icl>do(agt>person, obj>thing).@entry, ውሻ(icl>mammal).@def.@male)

ins('አባረረ' (icl>do(agt>person, obj>thing).@entry, ድንጋይ(icl>thing))

{/unl}



The first step of the algorithm is to generate natural language words from each universal word by considering their attributes using the morphological generator. The universal word ***አስቴር(icl>female person)*** has no attribute and will be changed to the noun ***አስቴር*** and the universal word ***ድንጋይ(icl>thing)*** will be changed to the word ***ድንጋይ*** because they have no attributes. The universal word has ***ውሻ(icl>mammal).@def.@male*** two attributes ***.@def*** and ***@male***. The morphemes used to indicate definiteness in Amharic language is **-** ***ው*** and to indicate male gender is **-** ***ኡ*** which are found from the morphology of the Amharic language. The grammar rules show that the definiteness indicator comes before the gender indicator. So, the resulting word will be ***ውሻው*** which will be used as the object of the sentence.

The stem of the main verb of the sentence is ***አባረር*** and the subject of the verb is female person. So, to indicate the subject is a female person, we use the suffix **-** ***ች*** after the verb of a sentence in Amharic language. The object of the verb is definite and male in gender. The suffix used to indicate definiteness and masculine gender is **-** ***ው*** when the last letter of the stem is consonant. The suffix that indicates the attributes of the subject comes before the suffixes that indicate the attributes of the subject in the grammar of Amharic sentences. So, the main verb of the sentence ***አባረር*** will become ***አባረረችው*** using the rules for the grammar and the morphology of the Amharic language.

The other grammar rules to be considered are when we indicate the action is done using an instrument in Amharic language; we use the prefix ***በ*** **-** before the instrument used. The last thing is we have to use the suffix **-** ***ን*** after the object when we indicate the object of the sentence is the receiver of the action.

So, the resulting words will be ***አባረረችው***, ***ውሻውን***, ***አስቴር*** and ***በድንጋይ*** and their lexical category will be verb, object, subject and instrument of the action respectively. The order of the sentence in Amharic language is SOV (subject-verb-object) and the instrument comes before the main verb. Finally, the resulting sentence will be ***አስቴር ውሻውን በድንጋይ አባረረችው*** and this sentence is equivalent with the meaning of the UNL representation given as an input.

We can see the DeConversion process with the example we see in the previous components which are the UNL EnConversion and the UNL analysis. The input for the DeConversion

process will be the resulting UNL expression from the UNL analysis process. The UNL expression is

{unl}

obj('ወደመ'(obj>thing).@entry,@past, :01)

pos:01('መጽሃፍ'(icl>thing).@entry.@def.@plu, ኦልማዝ(icl>female person)

{/unl}

Clearly, we can see that the main verb of the sentence is *ወደመ* in the past form and the object is *መጽሃፍ*. The owner of the object *መጽሃፍ* is *ኦልማዝ* and to indicate the ownership in Amharic language, we use the prefix - የ before the noun which is the owner. So, from the morphological generator component, we get the words *ወደመ· because the noun is plural, መጽሃፍቶች to indicate* the plurality and definiteness and የኦልማዝ to indicate ownership from the morphological generator. Using the grammar rules, we arrange these words to get the correct sentence which is የኦልማዝ *መጽሃፍቶች ወደመ·* which is not present in the input sentences and can be used as a summary for the input sentences.

4.8 Summary

In this chapter, we described the main components of the automatic Amharic text summarizer using abstractive approach. The main components of the summarizer are preprocessing module, clustering module, the UNL EnConverter, the UNL analyzer and the UNL DeConverter. The main reason we use the preprocessing module is to make it easy for the second module which is a clustering module. The preprocessing module has different steps like normalization, stop-word removal and stemming that makes the clustering process easy and effective. The clustering module is used to group sentences which have similar or related meaning in a single cluster. This clustering process will make the succeeding processes which are based on the UNL representation of sentences. If sentences with similar or related meaning are grouped together, it will be easy for the UNL analyzer component to get a single UNL representation that could be used as a representative for the cluster.

The UNL EnConverter and DeConverter are components that used to convert between the natural language sentences and the UNL representation of the natural language sentences. The EnConverter component is used to convert from natural language sentence in to the UNL representation using the grammar rules of Amharic language and the morphological analyzer for the Amharic language. The DeConverter component is used to convert the UNL representation of sentences into natural language sentences using the grammar rules of Amharic language and the morphological generator for the Amharic language.

Since Amharic is one of the morphologically complex languages, we have used many components like the grammar rules and the morphological analyzer and generator in order to use the UNL expression to represent the semantic meaning of natural language sentences.

Chapter Five: Experiment

4.1 Introduction

This chapter discusses about the experiment carried out to evaluate the validity of our proposed system. The datasets that are used in the experiment and the environment in which the proposed system is implemented are also discussed. Some of the components in the proposed system are not fully implemented for Amharic language and we try to use manual results for some test data.

As we have mentioned earlier, this thesis work is the first attempt of developing an Amharic text summarizer using abstractive approach, we have conducted a different type of experiment to the previous works of Amharic text summarization. The previous works of Amharic text summarization, as they are based on extractive approach, held their experiment by preparing manual summaries and checking the system summaries against the manual summaries.

Usually, researches that are done on text summarization systems which are of extractive approach used precision and recall to evaluate their work. Precision and recall are measures that are used to evaluate the system summary against the summary which is prepared manually [4, 9]. Since, their aim is at selecting top ranking sentences from the input document, precision and recall are effective to be used as evaluation metrics because of high tendency of having ideal summary using manually. So, if we have an ideal summary, we can evaluate our system against the ideal summary.

The summary systems which are of abstractive approach should not be evaluated by manual summaries as they have higher degree of subjectivity when compared to summary systems which follow extractive approach. Since these systems generate new sentences that may not be present in the input document, it is difficult to have an ideal summary in which the system summary is evaluated against. So, we have used different evaluation metrics in order to evaluate our system. These metrics are grammar and idea of the summary. Evaluators will evaluate the quality of summaries produced by the system using the above metrics.

4.2 Development Tools

Java programming and python are used to develop the prototype. Some components of the proposed system, where morphology of a language is needed, are implemented using python. We have used the morphological analyzer developed by Michael Gasser with a little modification as the system does not fully support all Amharic words. MySQL is also used as a database for different purpose.

4.3 Test Data

A total of 20 documents are used for evaluation purpose. We have chosen the test documents based on their number of sentences and their topic. The topic is considered because of its

contribution to the expected number of sentences in the summary. Input documents which have different main topics tend to have more number of summary sentences and this may result in more variation from the expectation of the evaluators. This is because, as it can be seen in chapter four, the proposed system has a clustering component which clusters sentences with more related idea in the same cluster. If we have an input document which has many topics covered, it may result in many number of clusters and the number of summary sentences will be higher.

The test documents are found from the internet. Some are from Facebook and some are from online news media like reporter.

4.4 Test Measures

The test measure we used for evaluating the proposed system is using users who have adequate capacity on Amharic language. There are a total of 3 evaluators who have evaluated the system using 20 input documents. The evaluators used the metrics mentioned above which are grammar and idea of the summary and put the value of each metrics based on the levels given.

Grammar test

Grammar is a rule in which a natural language sentences are legally constructed. A grammatically correct natural language sentence is one which is well formed, contains correct spellings of words, maintains the correct order of word categories and conveys intended meaning. We have used four levels to describe the grammar of sentences. The levels of a grammar test together with their score and description are described in table5.1.

Table 5.1 scales of grammar test

Level	Score	Description
Perfect	4	Good grammar
Good	3	Understandable but with grammar errors
Fair	2	Hard to understand but easy to guess
Bad	1	Not understandable

Idea test

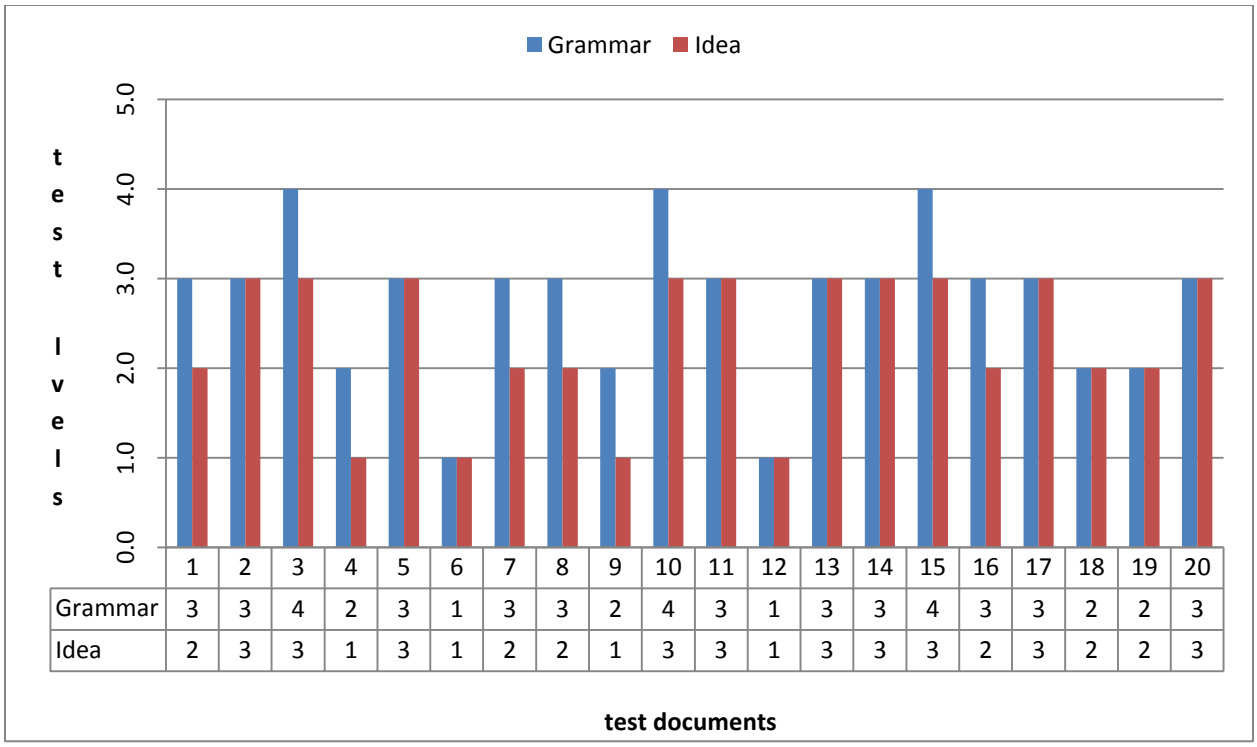
The idea of a summary refers to the central idea of the input document. At the beginning, the summary of a text is needed to have the main idea of a given text which small and precise. So, this test is intended to check whether the central idea or main idea of the input document is still presented in the summary or not. For this test, we have used three levels. The levels are described in table 5.2

Table 5.2 scales of an idea test

Level	Score	Description
Good	3	It has the central idea of the input document
Fair	2	Some portion of the central idea of a document is presented
Bad	1	The central idea of the input document is not presented

4.5 Results and Discussion

The human evaluators used the levels provided for each test measure to test the performance of the proposed system. The result of the grammar and idea test for the input documents given by each of the evaluators is given in figure 5.1, 5.2 and 5.3 respectively.

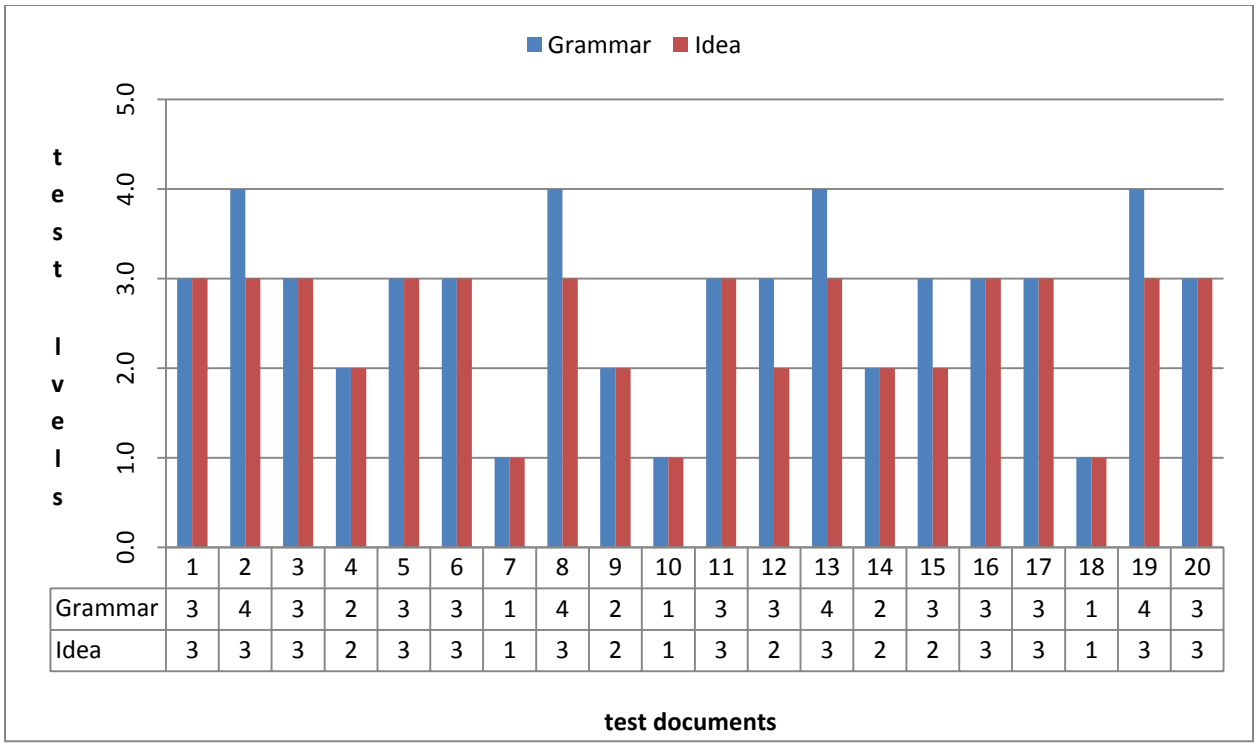


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Grammar	3	3	4	2	3	1	3	3	2	4	3	1	3	3	4	3	3	2	2	3
Idea	2	3	3	1	3	1	2	2	1	3	3	1	3	3	3	2	3	2	2	3

Fig 5.1 experimental result for grammar and idea test of evaluator 1

As it can be seen from the graph in figure 5.1, 11 document summaries which are 55% of the test data are good in grammar and 3 document summaries which are 15% of the test data are perfect. 4 document summaries which are 20% of the input data are fair in grammar and the rest 2 document summaries which are 10% of the input data have a bad grammar.

And when we come to the idea test, 10 document summaries which are 50% of the test data are given a score of good which means they represent the central idea of the input document whereas 6 document summaries which are 30% of the test data are fair in representing the central idea of the input documents. The rest 4 document summaries which are 20% of the test data did not represent the central idea of the input document.

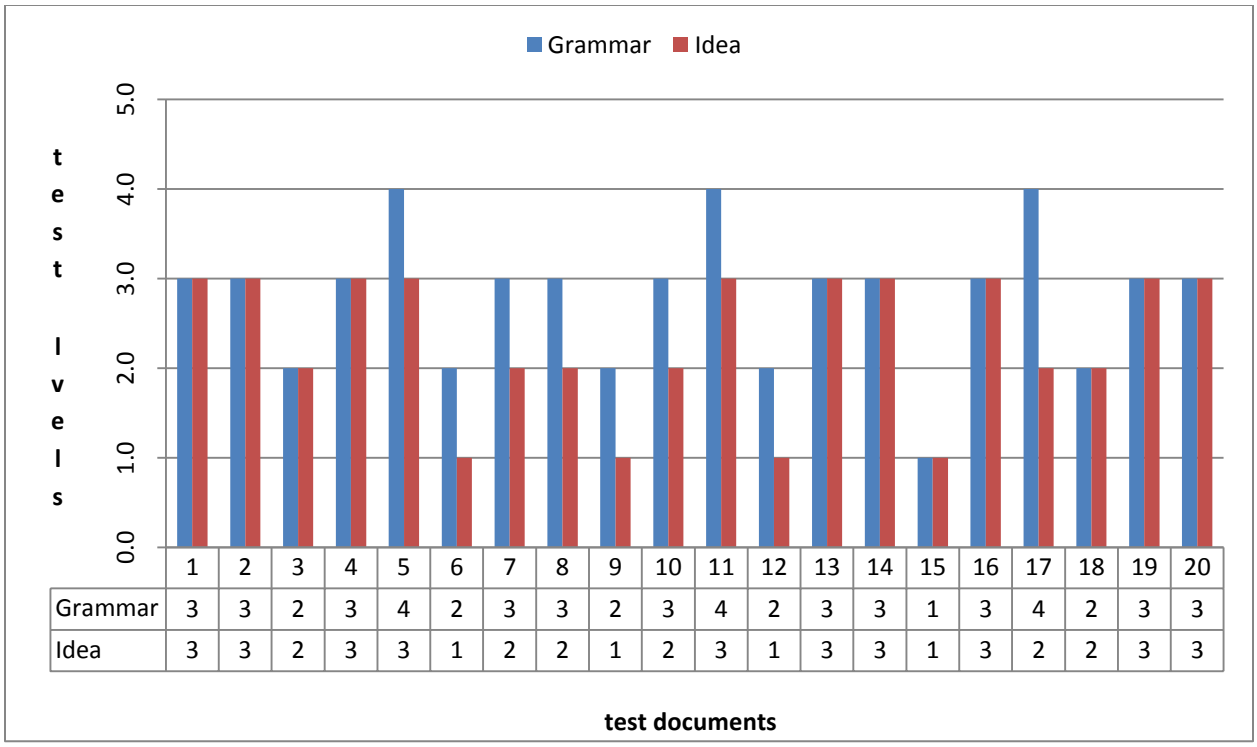


	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Grammar	3	4	3	2	3	3	1	4	2	1	3	3	4	2	3	3	3	1	4	3
Idea	3	3	3	2	3	3	1	3	2	1	3	2	3	2	2	3	3	1	3	3

Fig 5.2 experimental result for grammar and idea test of evaluator 2

As it can be seen from the graph in figure 5.2, 10 document summaries which are 50% of the test data are good in grammar and 4 document summaries which are 20% of the test data are perfect. 3 document summaries which are 15% of the input data are fair in grammar and the rest 3 document summaries which are 15% of the input data have a bad grammar.

And when we come to the idea test, 12 document summaries which are 60% of the test data are given a score of good which means they represent the central idea of the input document whereas 5 document summaries which are 25% of the test data are fair in representing the central idea of the input documents. The rest 4 document summaries which are 20% of the test data did not represent the central idea of the input document.



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Grammar	3	3	2	3	4	2	3	3	2	3	4	2	3	3	1	3	4	2	3	3
Idea	3	3	2	3	3	1	2	2	1	2	3	1	3	3	1	3	2	2	3	3

Fig 5.3 experimental result for grammar and idea test of evaluator 3

As it can be seen from the graph in figure 5.3, 11 document summaries which are 55% of the test data are good in grammar and 3 document summaries which are 15% of the test data are perfect. 5 document summaries which are 25% of the input data are fair in grammar and the rest 1 document summary which is 5% of the input data have a bad grammar.

And when we come to the idea test, 10 document summaries which are 50% of the test data are given a score of good which means they represent the central idea of the input document whereas 6 document summaries which are 30% of the test data are fair in representing the central idea of the input documents. The rest 4 document summaries which are 20% of the test data did not represent the central idea of the input document.

4.6 Summary

In developing text summarization systems, the evaluation process is challenging since there is no standard way to evaluate the performance of the system. When human evaluators are used to evaluate the performance of the system, since different evaluators may have different summaries, it has a high degree of subjectivity. But, in our case human evaluators are not expected to prepare manual summaries rather they are expected to read the input documents and judge the summaries based on different metrics like the grammar of the generated summary and whether the summary is valuable in representing the main idea of the input document or not.

We have used human evaluators to test the performance of our proposed system and the results are promising as it is the first attempt to do the summarization system in abstractive approach. When the components of our proposed system are fully implemented for Amharic language, the system will have a better performance.

Chapter Six: Conclusion and Future Work

5.1 Conclusion

As there is a rapid growth in information usage online for different languages, text summarization systems are needed for users to get the main ideas that are covered by that huge amount of data on a specific field. Amharic is one of the languages that enjoy a good number of users online and to access different information on particular area, the need of text summarization systems arises.

Text summary can be defined as a text that is produced from one or more texts that contain a significant portion of the information in the original text. The user can determine whether the complete document is relevant or not by reading the summary since the summary conveys the main ideas of the whole document. Automatic text summarization is the process of automatically creating a compressed version of a given text that provides useful information to the reader.

There are two broad categories of text summarization approaches. These are extractive and abstractive text summarization systems. Text summarization systems which are of extractive approach produce a text summary by selecting summary sentences from the input sentences using different techniques. Abstractive text summarization systems produce summary sentences that may not be presented in the input document. The abstractive approach for text summarization is done using the semantic representation of input sentences to produce new summary sentences that could be representatives for the meaning of different sentences.

In this thesis, we explain the development of Amharic text summarizer that follows abstractive approach for text summarization. The design of the summarizer is consists of five major components. The components are the preprocessing module, the sentence clustering module, the UNL EnConverter, the UNL analyzer and the UNL DeConverter.

The preprocessing module undergoes different stages like normalization, stop-word removal and stemming. The main purpose of the preprocessing module is to provide a good input for the clustering module. The different stages of the preprocessing module make each word of the input sentences in their root form. This makes the next step which is clustering of sentences more effective. The clustering module group sentences with similar meaning in a single cluster using the Wordnet dictionary to determine the relation in meaning for different words.

The other components of the proposed system are related with the semantic representation of natural language sentences using the Universal Networking Language (UNL) expression. The EnConverter is used to convert natural language sentences into UNL expression and the DeConverter performs the conversion from UNL expression into natural language sentences which is the reverse operation of the EnConverter module. The UNL analyzer is used to analyze

UNL expressions of many sentences which are in the same cluster and come up with representative UNL expression that could be used as the summary for the cluster.

The difficulty of NLP applications that involve Amharic in general and this research in particular arises from the complexity of the morphology of the language. In Amharic language, each lexical category of a word can be inflected to many aspects of sentences. It is also possible to indicate a full sentence using a single word when the word is inflected for the subject and object of a sentence using different morphemes. The other property of Amharic morphology is the unpredictability of the morphological patterns. For each pattern of morphological generation, there are many cases that should be considered as an exception. These properties make the development of natural language applications for Amharic language difficult and challenging.

For evaluation of the proposed system, a dataset consists of 20 Amharic documents was used. Five human evaluators are also used to measure the validity of the summary sentences using different criteria. The criterion used by the human evaluators is the grammar of the summary sentences and whether the main idea of the original document is presented in the summary or not. Human evaluators are not expected to produce manual summaries rather they give different values for the grammar of the summary sentence and for the main idea of the input document.

5.2 Future work

The following lists are some research directions that could be done on automatic Amharic text summarization

- The rules that are used in the morphology and grammar of the language can consider generic rules to apply them to more complex sentences
- The rules of UNL EnConversion and UNL DeConversion could be enriched to make the process more effective
- The machine learning approach can be considered for the processes of UNL related tasks.

Reference

[1] Asef Poormasoomi, Mohsen Kahani, Saeed Varasteh Yazdi and Hosssein Kamyar, “Context Based Persian Multi-Document Summarization (global view)”, International Conference on Asian Language Processing, 2011.

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

[3] HP. Luhn, “The Automatic Creation of Literature Abstracts”, IBM Journal of Research and Development, Vol. 2, Issue 2, pp. 159-165, April 1958.

[4] Melesse Tamiru, “Automatic Amharic Text summarization using latent semantic analysis”, Unpublished Master’s Thesis, Addis Ababa University, October, 2009.

[5] Elena Lloret, “Text Summarization: an Overview”, Deptartment Lenguajes y Sistemas Informaticos, Universidad de Alicante, Alicante, Spain

[6] Martin Hassel, "Resource Lean and Portable Automatic Text Summarization”, Unpublished Doctoral dissertation, Royal Institute of Technology, Stockholm, Sweden, 2007.

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

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

[9] Habtamu Demile, “Topic-Based Multi-Document Summarization for Amharic Text”, Unpublished Master’s Thesis, Addis Ababa University, February, 2014.

[10] Eyob Delele, “Topic-based Amharic Text Summarization”, Unpublished Master’s Thesis, Addis Ababa University, March 2011.

[11] Barzilay, R.: “Lexical Chains for Summarization”, M.Sc. thesis, Ben-Gurion University of the Negev, Unpublished, 1997.

[12] Makbule Gulcin Ozsoy, Ferda Nur Alpaslan and Ilyas Cicekli, „ Text summarization using latent semantic analysis“, Journal of Information Science published online 27 June 2011.

[13] Johanna Geiß, “Latent semantic sentence clustering for multi-document summarization”, Technical reports published by the University of Cambridge, 2011.

[14] Tessema Mindaye Mengistu: “Design and Implementation of Amharic Search Engine.” Msc Thesis, Addis Ababa University, Unpublished, Addis Ababa, 2007.

[15] M. en C. Yulia Nikolaevna Ledeneva: “Automatic language- independent detection of multi- word description for text summarization”, Laboratory of Natural Language and Processing of text, 2008.

[16] Ani Nenkova, Kathleen McKeown, “Automatic Summarization”, Foundation and Trends in Information Retrieval, Vol. 5, pp. 103-233, 2011.

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

[18] J. Kupiec, J. Pedersen, and F. Chen, “A trainable document summarizer,” in Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 68–73, 1995.

[19] J. M. Conroy and D. P. O’Leary, “Text summarization via hidden Markov models,” in Proceedings of the Annual International ACM SIGIR Conference.

[20] R. Barzilay and N. Elhadad, “Sentence alignment for monolingual comparable corpora,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing, pp. 25–32, 2003.

[21] T. Copeck and S. Szpakowicz, “Leveraging pyramids,” in Proceedings of the Document Understanding Conference, 2005.

[22] H. Jing and K. McKeown, “The Decomposition of Human-Written Summary Sentences,” in Proceedings of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 129-136.

[23] H. Jing, “Sentence reduction for automatic text summarization,” in Proceedings of the Conference on Applied Natural Language Processing, pp. 310-315, 2000.

[24] H. Jing and K. R. McKeown, “Cut and paste based text summarization,” in Proceedings of the North American chapter of the Association for Computational Linguistics Conference, pp. 178–185, 2000.

[25] R. Barzilay and K. R. McKeown, “Sentence Fusion for Multi-document News Summarization,” Computational Linguistics, vol. 31, no. 3, pp. 297–328, 2005.

[26] Vishal Gupta and Gurpreet Singh Lehal: “A Survey of Text Summarization Extractive Techniques” Journal of Emerging Technologies in Web Intelligence, Vol. 2, No. 3, 2010.

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

[28] Karel Jezek and Josef Steinberger: “Automatic Text Summarization (The state of the art 2007 and new challenges, Department of Informatics and computational techniques, WEU - Západoèeská University, Pilsen, 2008.

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

[30] Marvin L. Bender, Head W. Sydeny, and Roger Cowley. *The Ethiopian Writing System*, In Bender et al (Eds.) Language in Ethiopia. London, Oxford University press, 1976.

[31] Martha Y.T., Morphology-Based Language Modeling for Amharic, PhD dissertation, Hamburg University, Unpublished, Hamburg, 2010.

[32] Baye Yimam, የአማርኛ ሰዋሰው, EMPDE, Addis Ababa, 2nd edition, 2000EC.

[33] Bender M.L., Bowen J.D., Cooper R.L., and Ferguson C.A., Languages in Ethiopia, Oxford Univ. Press, London, 1976.

[34] Titov E. G., The Modern Amharic Language, NAUKA Publishing House, Moscow, 1976, 47-48.

[35] S. Alansary, M. Nagi and N. Adly, Machine Translation Using the Universal Networking Language (UNL), Alexandria University, Alexandria, Egypt

[36] H. Uchida and M. Zhu, “UNL for Providing Knowledge Infrastructure,” in Proceedings of the Semantic Computing Workshop (SeC2005), Chiba, Japan, 2005.

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

[38] Barzilay, R., Elhadad, M., Using Lexical Chains for Text Summarization. In Proceedings of the ACL/EACL’97 Workshop on Intelligent Scalable Text Summarization, Madrid, Spain, 1997, pp. 10–17.

[39] Radev R., Blair-goldensohn S., Zhang Z., Experiments in Single and Multi-document Summarization using MEAD, in First Document Understanding Conference, New Orleans, LA, 2001.

[40] Lin C. Y., Hovy, E. Automated Multi-document Summarization in NeATS. In Proceedings of the Human Language Technology (HLT) Conference, San Diego, CA, 2001.

[41] Svore K, Vanderwende L, Burges C, Enhancing Single-Document Summarization by Combining RankNet and Third-Party Sources, In Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 448-457, 2007.

[42] Daum'e III H, Echihabi A., Marcu D., et al. GLEANS: A Generator of Logicl Extracts and Abstracts for Nice Summaries, In Workshop on Text Summarization (In Conjunction with the ACL 2002 and including the DARPA/NIST sponsored DUC 2002 Meeting on Text Summarization), Philadelphia, 2002.

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

[44] Teferi Andargie, "The Application of Machine Learning Technique (Naïve-Bayes) for Automatic Text summarization (The Case of Amharic News Texts)", Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, 2005.

[45] Helen Adane, "Text Summarization on Amharic Legal Judgments", Master‟s Thesis, Addis Ababa University, Addis Ababa, Ethiopia, Unpublished, 2006.

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

[47] Radev D., Weiguo F., Zhang Z., WebInEssence: A personalized web-based multi-document summarization and recommendation system, In NAACL Workshop on Automatic Summarization, Pittsburg, 2001.

[48] Jos_e Abracos and Gabriel Pereira Lopes: Statistical methods for retrieving most significant paragraphs in newspaper articles, 1997.

[49] Dimitrios Galanis, "Automatic Generation of Natural Language Summaries", Unpublished PhD Dissertation, Athens University of Economics and Business, 2012.

[50] Michael Gasser. Hornmorpho2.1 user's guide, 2010. URL http://www.cs.indiana.edu/~gasser/L3/horn2.1.pdf. 63, 115

Annexes

Annex A: Description of UNL relations

UNL relation	Description	Constituent elements	Examples
1. Agt	defines a thingthat initiate an action	Agent, intransitive verbs (intransitive verb semanticallyh avean agent subject) like ‘sleep’,‘snore’ ,‘cough’,‘run’, etc.	John slept … agt(slept, John) John killed Mary. agt(killed, John) … arrival ofJohn … agt(arrival, John) …playbyShakespeare agt(play, Shakespeare)
2. And	defines a partner to have conjunctiverelation	Conjunction	… easilyand quickly… and(quickly, easily) … singingand dancing … and(dance(agt>person),sing(agt>person))
3. Aoj	defines a thingthat is in a stateor has an attribute	Stative verbs like ‘believe’, ‘understand’,‘know’, ‘have’,‘possess’	John believes in Mary. aoj(believes, John)

		, ‘dislike’, ‘love’, ‘like’,‘contain’, ‘include’, ‘involve’, etc.	John knows Mary. aoj(knows, John) John loves Mary. aoj(loves, John)
		aoj (general attribute)	John issad. aoj(sad, John) John lookssad …. aoj(sad, John)
4. Bas	defines a thingused as the basis (standard)of comparison	Basis	... morethan seven. bas(more(aoj>thing,bas>thing), 7) … morethan Jack. bas(more(icl>how,bas>thing), Jack(iof>person))
5. Ben	defines anindirectly related beneficiaryor victimof an event or state	Beneficiary	To give one’s life for one’scountry. ben(give(agt>thing, gol>thing,obj>thing), country(icl>region)) It isgood forJohn to … ben(good(aoj>thing),

			John(iof>person))
6. Cag	Defines athingnot in focus that initiates an implicitevent that is donein parallel	Co-agent	… to walk with John. cag(walk(agt>volitional thing), John(iof>person)) To livewith …aunt. cag(live(agt>volitional thing), aunt(icl>person))
7. Cao	Defines a thingnot in focus that is in a parallel state	Co-thingwith attribute	... bewithyou … cao(exist(aoj>thing),you)
8. cnt	defines the content ofa concept	Content	TheInternet:anamalgamation… cnt(Internet(icl>communication network), amalgamation(icl>harmony)) A language generator "deconverter"… cnt(language generator, deconverter.@double_quote)
9. cob	defines a thingthat is directly affected byan implicitevent donein parallel oran implicitstate in parallel	Affected co-thing	… dead with Mary. cob(die(obj>livingthing), Mary(iof>person))
10. con	defines a non- focused event orstatethat conditions a focused eventor state	Condition	Ifyou aretired, wewillgostraight home. aoj:01(tired(aoj>thing),you) con(go(icl>move(agt>thing, gol>place,src>place)), :01)
11. coo	defines aconcurrent event or state fora focused event or state	While	… worked while … talked coo(worked, talked)

12. dur	defines a period oftime duringwhich an event occurs ora state exists	During	John worked during… dur(worked, meeting)
13. equ	defines an equivalent concept	Equivalent	Thedeconverter(alanguage generator)… equ(deconverter, language generator.@parenthesis)
14. fmt	defines a range between two things	Range/from-to	… from a to z. fmt(z(icl>letter), a(icl>letter)) … from Osakato New York. fmt(New York(iof>city), Osaka(iof>city))
15. frm	defines an initial state of athingora thinginitially associated with the focusedthing	Origin	A visitor from Japan … frm(visitor(icl>person), Japan(iof>country))
16. gol	defines a final stateof object orathing finally associated with the object of an event	Final state ofverbs of changelike ‘give’, ‘send’, etc.	… gave… to Mary. gol(gave, Mary) … sent … to Mary. gol(sent, Mary)
17. icl	defines an upper concept oramore general concept	Included/akind of	A bird is a (kind of)animal. icl(bird(icl>animal), animal(icl>livingthing))
18. ins	defines an instrumentto	Instrument	… look at stars througha telescope. ins(look(agt>thing,obj>thing),

	carryoutan event		telescope(icl>optical instrument)) … writewith a pencil. ins(write(agt>thing,obj>thing), pencil(icl>stationery))
19. int	defines all common instances to havewith a partner concept	Intersection	…. an intersection oftableware and cookware… int(tableware(icl>tool), cookware(icl>tool))
20. iof	defines a class concept that an instance belongs to	An instanceof	Tokyo is acityin Japan. iof(Tokyo(iof>city), cityin Japan)
21. man	defines a way to carryout an event or the characteristics ofastate	Manner	… move quickly. man(move(agt<thing, gol>place,src>place), quickly) … often visit ... man(visit(agt>thing, obj>thing), often)
22. met	defines a means to carryoutan event	Method/ means	… solve… with dynamics. met(solve(icl>resolve(agt>thing,obj>thing)), dynamics(icl>science)) … separate… bycutting … met(separate(agt>thing,obj>thing,src>thing), cut(agt>thing,obj>thing, opl>thing))
23. mod	defines a	Modification	Thewholestory…

	thingthat restricts a focusedthing		mod(story(icl>tale), whole(mod<thing)) A masterplan …. mod(plan(icl>idea),master(mod<thing))
24. nam	defines a name ofa thing	Name	... son “Hikari” nam(son(icl>relative),Hikari)
25. obj	defines a thingin focus that is directly affected byan event or state	Un-Accusative verbs like ‘die’, ‘fall’, ‘melt’, etc.	John died. obj(died, John) The snow melts. obj(melts,snow)
		obj (direct object)* Accusative	John killed Mary. obj(killed, Mary) John knows Mary. obj(knows, Mary) John loves Mary. obj(loves, Mary)
		obj (indirect object of mono-transitive verbs like ‘depend’,‘believe’, ‘laugh’, etc.)	… depends on Mary. obj(depends, Mary) … believes in Mary. obj(believes, Mary)

		object	… construction ofthe building... obj(construction, building) … interest in Physics … obj(interest, Physics) … visitto London. obj(visit, London)
26. Opl	defines a placein focus affected byan event	Affected place	… pat … on shoulder …. opl(pat(icl>touch(agt>thing, obj>thing,opl>thing)), shoulder(pof>trunk)) … cut … in middle … opl(cut(agt>thing,obj>thing, opl>thing), middle(icl>place))
27. Or	defines a partner to have disjunctive relation to	Disjunction	… stayor leave… or(leave(agt>thing,obj>place), stay(icl>remain(agt>thing))) Is It red or blue? or(blue(icl>color), red(icl>color))
28. Per	defines a basisor unit ofproportion, rate or distribution	Proportion/ rate/ distribution	… hours a day. per(hour(icl>period), day(icl>period))
29. Plc	defines a placewhere an event occurs, ora stateis true, orathing exists	Place	MadeinItaly. plc(made,Italy)

30. plf	Theplace where an event begins orastate becomes true	Initial placeof verbs of motion like ‘go’, ‘travel’, ‘walk’, ‘come’,etc.	John came from German. plf(came,German)
31. plt	defines a placewhere an event ends orastatethat becomes false	Final place	… to travel to Panama. plt(travel(agt>volitional thing), Panama (iof>city))
32. pof	indicate a concept of which a focusedthing is a part	Part of	Thepreambleof a document … pof(preamble(icl>information), document(icl>information)) … the initials of Machine Translation … of(initial(icl>letter), machine translation)
33. pos	defines the possessor ofa thing	Possessor	John’s dog. pos(dog(icl>aminal), John(iof>person)) Mybook. pos(book(icl>document),I)
34. ptn	defines an indispensable non-focused initiator of an action	Partner	… competewith John … ptn(compete(agt>thing, ptn>thing), John(iof>person)) … share… with thepoor. ptn(share(icl>divide(agt>thing, obj>thing)), poor(icl>person))
35. pur	defines the purposeor	Purpose	*John works for money.* *agt(work(icl>do),*

	objectiveof an agent ofan event or the purposeof a thingthat exists		*John(iof>person))* *pur(work(icl>do),money)* … budgetforresearch … pur(budget(icl>expense), research(icl>study))
36. qua	defines the quantityof a thingor unit	Quantity	Two cups of coffee… qua(cup(icl>tableware),2)) qua(coffee(icl>beverage), cup(icl>tableware)) … manykilograms … qua(kilogram(icl>unit), many(qua<thing)) … two dogs … qua(dog(icl>animal), 2)
37. rsn	defines a reason why an event ora statehappens	Reason	*Hegoes ... becauseof ...* *illness.* rsn(go(icl>do), illness(icl>thing)) … known for… beauty. rsn(known(aoj>thing), beauty(icl>abstract thing))
38. scn	defines a scene where an event occurs, or state is true, or a thing exists	Scene	*... appear on a program.* scn(appear(icl>occur), program(icl>thing))
39. seq	defines a prior event or stateof a focused event orstate	After	John worked and left … seq(left, worked)

40. shd	defines a number, a mark or a thingthat shows the position ofa sentence,a paragraph or a chapter in a document ora book	Sentencehead	Chapter2 Relation shd(relation(icl>sate), chapter(pof>book)) mod(chapter(pof>book),2)
41. src	defines the initial state of an object or thinginitially associated with the object of an event	Initial state ofverbs of changelike ‘take’, ‘retrieve’, etc.	*... changed from red ...* src(change(icl>occur), red(aoj>thing))
42. tim	defines the time an event occurs ora stateis true	When	… cameyesterday… tim(came,yesterday)
43. tmf	defines the time an event starts or a statebecomes true	Sincewhen	… worked sinceearly… tmf(worked, early)
44. tmt	defines the time an event ends or astate becomes false	Untilwhen	… worked until late … tmt(worked, late)
45. to	defines a final stateof a thingor a final thing (destination) associated with the focusedthing	Destination	*... train for London ...* to(train(icl>thing), London(icl>city))

46. via	Defines an intermediate placeor state of an event	An intermediate placeor State	*... goes to ... via New York.* via(go(icl>do), New York(icl>place))

Annex B: Description of UNL attributes

Concept	Attributed as
Logicality of UW	@transitive, @symmetric, @identifiable, @disjointed
Time with respect to the speaker	@past, @present, @future
Speaker's view on aspects of event	@begin, @complete, @continue, @custom, @end, @experience, @progress, @repeat, @state @just, @soon, @yet
Speaker's view of reference to concepts	@generic, @def, @indef, @not, @ordinal
Speaker's emphasis, focus and topic	@contrast, @emphasis, @entry, @qfocus, @theme, @title, @topic
Speaker's attitudes	@affirmative, @confirmation, @exclamation, @humility, @imperative, @interrogative, @invitation, @polite, @request, @respect, @vocative
Speaker's feelings and judgments	Attributes to represent ability: @ability Attributes to represent beneficially: @get-benefit, @give-benefit Attributes to represent conclusion: @conclusion, @consequence Attributes to represent condition: @sufficient Attributes to represent consent/dissent: @consent, @dissent, @grant, @grant-not Attributes to represent expectation: @although, @discontented, @expectation, @wish Attributes to represent intention: @insistence, @intention, @want, @will, @need, @obligation, @obligation-not, @should, @unavoidable Attributes to represent possibility: @certain, @inevitable, @may, @possible, @probable, @rare, @unreal Attributes to represent emotion: @admire, @blame, @contempt, @regret, @surprised, @troublesome

Concept	Attributed as
Convention description	@passive, @pl, @angle_bracket, @brace, @double_parenthesis, @double_quote, @parenthesis, @single_quote, @square_bracket

Annex C: List of short words and their expanded form

ት/ቤት	ትምህርት ቤት
ት/ርት	ትምህርት
ት/ክፍል	ትምህርት ክፍል
ሃ/አለቃ	ሀምሳ አለቃ
ሃ/ስላሴ	ሀይለ ስላሴ
ደ/ዘይት	ደብረ ዘይት
ደ/ታቦር	ደብረ ታቦር
መ/ር	መምህር
መ/ቤት	መስሪያ ቤት
መ/አለቃ	መቶ አለቃ
ክ/ከተማ	ክፍለ ከተማ
ክ/ሀገር	ክፍለ ሀገር
ወ/ር	ወታደር
ወ/ሮ	ወይዘሮ
ወ/ሪት	ወይዘሪት
ወ/ስላሴ	ወልደ ስላሴ
ፍ/ስላሴ	ፍቅረ ስላሴ
ፍ/ቤት	ፍርድ ቤት
ጽ/ቤት	ጽህፈት ቤት
ሲ/ር	ሲስተር
ፕ/ር	ፕሮፌሰር
ጠ/ሚኒስትር	ጠቅላይ ሚኒስትር
ዶ/ር	ዶክተር
ገ/ገዮርጊስ	ገብረ ገዮርጊስ
ቤ/ክርስትያን	ቤተ ክርስትያን
ም/ስራ	ምክትል ስራ
ም/ቤት	ምክር ቤት

ተ/ሃይማኖት	ተክለ ሃይማኖት
ሚ/ር	ሚኒስትር
ኮ/ል	ኮለኔል
ሜ/ጀነራል	ሜጀር ጀነራል
ብ/ጀነራል	ብርጋዴር ጀነራል
ሌ/ኮለኔል	ሌተናል ኮለኔል
ሊ/መንበር	ሊቀ መንበር
አ/አ	አዲስ አበባ
ር/መምህር	ርእሰ መምህር
ፕ/ት	ፕሬዝዳንት
ዓ.ም	አመተ ምህረት
ዶ.ር	ዶክተር

Annex D: List of stop-words for Amharic Language

ሁሉ
ሁሉም
ኋላ
ሁኔታ
ሆነ
ሆኑ
ሆኖም
ሁል
ሁሉንም
ላይ
ሌላ
ሌሎች
ልዩ
መሆኑ
ማለት
ማለቱ
መካከል
የሚገኙ
የሚገኝ
ማድረግ
ማን
ማንም
ሰሞኑን
ሲሆን
ሲል
ሲሉ
ስለ
ቢቢሲ
ቢሆን
ብለዋል
ብቻ
ብዛት
ብዙ
ቦታ
በርካታ
በሰሞኑ
በታች
በኋላ
በኩል
በውስጥ
በጣም
ብቻ
በተለይ
በተመለከተ
በተመሳሳይ
የተለያየ
የተለያዩ
ተባለ
ተገለጸ
ተገልጿል
ተጨማሪ
ተከናውኗል
ችግር
ታች
ትናንት
ነበረች
ነበሩ
ነበረ
ነው
ነይ
ነገር
ነገሮች
ናት
ናቸው
አሁን
አለ
አስታወቀ
አስታውቀዋል
አስታውሰዋል
እስካሁን
አሳሰበ
አሳስበዋል
አስፈላጊ
አስገነዘቡ
አስገንዝበዋል
አብራርተዋል
አበራርተው
አስረድተዋል
እስከ
እባክህ
እባክሽ
እባክዎ
አንድ
አንጻር
እስኪደርስ
እንኳ
እስከ
እዚሁ
እና
እንደ
እንደገለጹት
እንደተገለጸው
እንደተናገሩት
እንደአስረዱት
እንደገና
ወቅት
እንዲሁም
እንጂ
እዚህ
እዚያ
እያንዳንዱ
እያንዳንዳቸው
እያንዳንዷ
ከ