ADDIS ABABA UNIVERISTY

College of Natural Sciences

Afaan Oromo Text Summarization using Word Embedding

Lamesa Tashoma Fanache

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

November 4, 2020

Addis Ababa University

College of Natural Sciences

Lamesa Tashoma Fanache

Advisor: *Yaregal Assabie (PhD)*

This is to certify that the thesis prepared by *Lamesa Tashoma Fanache*, titled: *Afaan Oromo Text Summarization using Word Embedding* 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:	_____	_____	_____
Examiner:	_____	_____	_____

Abstract

Nowadays we are overloaded by information as technology is growing. This causes a problem to identify which information is reading worthy or not. To solve this problem, Automatic Text Summarization has emerged. It is a computer program that summarizes text by removing redundant information from the input text and produces a shorter non-redundant output text.

This study deals with development of a generic automatic text summarizer for Afaan Oromo text using word embedding. Language specific lexicons like stop words and stemmer are used to develop the summarizer. A graph-based PageRank is used to select the summary of worthy sentences out of the document. To measure the similarities between sentences cosine similarity is used. The data used in this work was collected from both secondary and primary sources. Afaan Oromo stop word list, suffix and other language specific lexicons are gathered from previous works done on Afaan Oromo. To develop a Word2Vec model we have gathered different Afaan Oromo texts from different sources like: Internet, organizations and individuals. For validation and testing 22 different newspaper topics are collected, from this, 13 of them have been used for validation while the rest 9 were employed for testing purpose.

The system has been evaluated based on three experimental scenarios and evaluation is made both subjectively and objectively. The subjective evaluation focuses on evaluation of the structure of the summary like informativeness of the summary, coherence, referential clarity, non-redundancy and grammar. In the objective evaluation we used metrics like precision, recall and F-measure. The result of subjective evaluation is *83.33%* informativeness, *78.8%* referential integrity and grammar, and *76.66%* structure and coherence. This work also achieved *0.527* precision, *0.422* recall and *0.468* F-measure by using the data we gathered. However, the overall performance of the summarizer outperformed by *0.648* precision, *0.626* recall and 0.058 F-measure when compared with the previous works by using the same data used in their work.

Keywords: Automatic Text Summarization, Word Embedding, Sentence Vector, PageRank, Cosine Similarity

Dedicated to:

My mom, **Mulunesh Kebede** and my father, **Tashoma Fanache**.

Acknowledgements

First of all, I would like to thank the Almighty God for His help in all aspects of my life. My deepest heartfelt gratitude also goes to my advisor **Dr. Yaregal Asabie (PhD)** for his critical comments on my work and helpful advice.

My heartfelt gratitude is extended to **Debela Tezera (PhD Candidate at Haramaya University), Chaltu Teshome** and all other my respondents.

Mom and Dad, this work is dedicated to you with great pleasure and honor. My Dad, **Tashoma Fanache**, thank you for your amazing help and follow-up since my childhood. I appreciate your advice through which you planted in my mind the passion for education and knowledge. My Mom **Mulunesh Kebede**, I do not know how I can express your endless love, which lets you drop out everything to look after and educate your children. You are always my hero. I look-up to you when I am in difficulties and you always have a solution.

My **brothers** and **sisters** you are always there for me and I am very thankful for that. You all are pushing me forward and showing me the way how to grow as a man. Finally, my thank goes to my friends who had directly or indirectly contributed in development of this thesis, without which it would not have reached this status. My friend **Desta Legesse,** I am thankful for your help in every aspect.

Table of Contents

List of Figures

List of Tables

List of Algorithms

Acronyms and Abbreviation

1.p	1st Person
1.p.pl	1st Person Plural
1.p.S	1st Person Singular
2.p	2nd Person
3.p	3rd Person
3.p.f	3rd Person Feminine
3.p.m	3rd Person Masculine
3.p.S	3rd Person Singular
ATS	Automatic Text Summarization
CBOW	Continuous Bag of Words
CNN	Convolutional Neural Network
GloVe	Global Vector
HITS	Hyper Induced Topic Search
HMM	Hidden Markov Model
HTML	Hyper Text Markup Language
IE	Information Extraction
IR	Information Retrieval
LSA	Latent Semantic Analysis
MMR	Maximal Marginal Relevance
NLP	Natural Language Processing
NLTK	Natural Language Toolkit
OTS	Open Text Summarizer
PLSA	Probabilistic Latent Semantic Analysis
RNN	Recurrent Neural Network
Seq2Seq	Sequence to Sequence
SVD	Singular Value Decomposition
TF-IDF	Term Frequency-Inverted Document Frequency
VSM	Vector Space Model

Chapter One: Introduction

1.1 Background

The exponential growth of online publishing provides users with a large amount of text on a great diversity of topics, which leads to redundancy and makes it difficult for users to find relevant information. To obtain information rich text contents, finding techniques that can generate a concise description of large documents has become urgent. In this scenario, automatic document summarization is considered to be an effective solution [1].

Automatic text summarization is part of machine learning, natural language processing (NLP) and data mining. It is becoming a popular research area while data grow and there is a demand to process it more efficiently. The aim is to find the core of the given text set and reduce the size while covering the key concepts and overall meaning and avoiding repetition [2].

Automatic text summarization is a technique where a program summarizes document or set of documents. A text is given to the program and the program returns a short and less redundant extract of the original text. Automatic text summarizer is vital towards reducing human effort through money and time. Automatic summarization has attracted attention both in the research community and commercially as a solution for reducing information overload and helping users to scan a large number of documents to identify documents of their interest [3].

Depending on the input document we can classify it to a single document and multi document text summarization. A single document text summarization produces summary of a single input document while multi-document summarization is an extraction of information from multiple texts written about the same topic. The resulting summary report allows individual users, such as professional information consumers, to quickly familiarize themselves with information contained in a large cluster of documents [4].

There are two main approaches to summarizing text documents: extractive and abstractive text summarization. Extractive text summarization involves the selection of phrases and sentences from the source document to make up the new summary. Techniques involve ranking the relevance of phrases in order to choose only those most relevant to the meaning of the source, while abstractive text summarization involves generating entirely new phrases

and sentences to capture the meaning of the source document. The abstractive approach is a more challenging approach, but is also the approach ultimately used by humans. Classical methods operate by selecting and compressing content from the source document [4].

Word embedding is one of the most popular representations of document vocabulary. It is capable of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. [5]. It is modern approach for representing text in natural language processing. Embedding algorithms like Word2Vec (word to vector) and GloVe (global vector) are key to the state-of-the-art results achieved by neural network models on natural language processing problems like machine translation, information retrieval (IR), text summarization, etc. Word embeddings work by using an algorithm to train a set of fixed length dense and continuous valued vectors based on a large corpus of text. Each word is represented by a point in the embedding space and these points are learned and moved around based on the words that surround the target word. There are two main training algorithms that can be used to learn the embedding from text; they are continuous bag of words (CBOW) and skip grams [6].

1.2 Motivation

The Oromo nation has a single common mother tongue, called the Oromo language or Afaan Oromo or Oromiffa [7]. It is the fourth most-widely spoken language in Africa as a mother tongue, next to Arabic, Swahili and Hausa. Today, Afaan Oromo is serving as an official language of Oromia regional state (which is the largest regional state among the current federal states of Ethiopia). Being an official language, a number of literatures, newspapers, magazines, education resources, official credentials and religious documents are published and available in the language.

Today, the improvement in modern technology is increasing the availability of digital information on the Internet, which is written by Afaan Oromo. Due to this, two basic problems are encountered, searching for relevant documents from this huge number of documents, and absorbing a large quantity of relevant information from these abundance documents [8]. In general, lack of active research on the automatic text summarization and a dramatic growth of electronic documents from time to time is motivating factors for this work to come up with a system that can minimize these problems.

1.3 Statement of the Problem

Automatic text summarization for different languages has been done so far by different researchers. There are many works for non-Ethiopian languages like English [9, 10, 11, 12, 13, 14], and Ethiopian languages like Amharic [15, 16, 17, 18]. However, because of the difference in the morphological structure of the languages we cannot directly apply these works for Afaan Oromo. Thus, attempts have been made to develop Afaan Oromo text summarization [8, 19, 20].

Girma Debele [19] tried to develop Afaan Oromo text summarization based on sentence selection methods. The researcher has used only two features to select a summary worthy sentence. The features are sentence position and term frequency. However, there are many other features which have to be taken into account in order to develop a good performance text summarization. These features are like cue phrases, name of events, numbers handling mechanism, sentence length etc.

Fiseha Berhanu [8] proposed a generic type of Afaan Oromo news text summarization. The researcher has tried to fill the research gap found in the work of Girma [19] by incorporating the above missed features. However, there is a sentence redundancy problem and lack of coherence and cohesion in this work.

Asefa Bayisa [20] has also tried to come up with a query based Afaan Oromo text summarizer. The author used two methods to extract important sentences from the document: The first one is one of the oldest and most extensively studied models of IR called vector space model (VSM). In VSM, both the sentences in a document and query are arranged in vectors. The angle between the document vector and the query vector is computed using cosine similarity measures. Hence, the sentences returned as an answer to a user query are those geometrically closest to the query according to the value obtained using the cosine similarity measure. Secondly, in an attempt to take document genre information into consideration; the position of each sentence in the document is used as additional features to compute the significance of a sentence. However, it is based on query terms given from the user (it is from the user perspective), not from document perspective. If the user has no idea about the contents of the document, it is difficult to select the keywords for the users.

Word embedding techniques are found to be very effective in finding contexts of words and relationship among words [6]. This helps to develop more accurate generic text summarization systems. In this work, we hypothesize that word embedding techniques may or may not overcome the problems for development of generic Afaan Oromo Text Summarization system.

1.4 Objectives

General Objective

The main objective of this research is developing automatic single document Afaan Oromo Text Summarization.

Specific Objectives

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

- Study linguistic characteristics of Afaan Oromo.
- Review related research works in the area of text summarization.
- Collect Afaan Oromo text and develop corpus.
- Select and customize or develop a summarization algorithm based on word embedding techniques for Afaan Oromo.
- Design a generic model for Afaan Oromo text summarizer.
- Develop a prototype summarizer that will serve as a model for Afaan Oromo text summarization.
- Evaluate the performance of the system.

1.5 Methods

In order to achieve the above specific objectives, the following methods will be followed.

Literature review

A literature review will be conducted to know Automatic text summarization in detail and to grasp a deep knowledge on Afaan Oromo and language technology.

Data collection

Data collection is the core point while dealing with natural language processing. We will gather two types of data categories to achieve our objective: the first one is the data needed to train our model. Since there is no pre-trained Afaan Oromo Word2Vec model we will gather

thousands of documents, from different domains. This data is used to develop the word embeddings. The second category is divided into three: the lexicon data, validation data and testing data. Lexicon data's like Afaan Oromo stop words and stems will be gathered from different sources. Validation and testing data will be gathered from different online Afaan Oromo news. We will collect news on different topics such as politics, technology, health, metrology, art, agriculture, education and sport.

Prototype Development

In order to evaluate the performance of the study, a prototype system will be developed for the Afaan Oromo text summarizer that can generate summary of the text.

Evaluation

Intrinsic evaluation method will be used in this work since it is the most widely used method for evaluation of text summarization. The evaluation will be undertaken in two ways subjective and objective ways. The subjective evaluation will focus on evaluation of the structure of the summary, such as summary referential integrity and non-redundancy, coherence and informativeness. On the other hand, the objective evaluation will use metrics like: Precision, Recall and F-measure.

1.6 Scope and Limitations

This research focuses on single document summarization for Afaan Oromo texts. It doesn't employ an abstractive summarization since it requires deep linguistic analysis and difficult to implement with current state of the art of the field.

1.7 Applications of Results

There are countless applications of automatic text summarization. Summarizations can increase the performance of traditional IR and IE systems (a summarization system coupled with Question-Answering (QA) system) [21].

Summarization can be applied to different domains:

- ✓ News summarization and Newswire generation
- ✓ Rich Site Summary (RSS) feed summarization
- ✓ Blog summarization
- ✓ Tweet summarization
- ✓ Web page summarization

- ✓ Email and email thread summarization
- ✓ Report summarization for business men, politicians, researchers, etc.
- ✓ Meeting summarization
- ✓ Biographical extracts
- ✓ Automatic extraction and generation of titles
- ✓ Domain-specific summarization (domains of medicine, chemistry, law, etc.)
- ✓ Opinion summarization, etc.

1.8 Organization of the Rest of the Thesis

The thesis is organized into seven Chapters including the current one. The second chapter presents the basic concepts and literatures on automatic text summarization. Chapter Three discusses Afaan Oromo. Chapter Four discusses about the related works which have been conducted in this area. Chapter Five presents the design of the system and activities carried out to implement the summarizer. Chapter Six presents the experimentation and evaluation and discussion of the result. Finally, Chapter Seven gives conclusions and future works.

Chapter Two: Literature Review

2.1 Text Summarization

Automatic Text Summarization (ATS) became a discipline in 1958 following Luhn's research into scientific text summarization [9]. It is a discipline of natural language processing (NLP) that aims to condense text documents. Condensing the text signifies producing a shortened version of the source document which contains the main points of the document. The process involves loss of information [22].

Textual information in the form of digital documents quickly accumulates to huge amounts of data. Most of this large volume of documents is unstructured: it is unrestricted text and has not been organized into traditional databases. Processing documents is, therefore, a perfunctory task, mostly due to the lack of standards. Consequently, it has become extremely difficult to implement automatic text analysis tasks. ATS, by condensing the text while maintaining relevant information, can help to process this ever-increasing and, difficult to handle, mass of information [23]. Text summarization is a way to condense a large amount of information into a concise form by the process of selection of important information and discarding unimportant and redundant information [22].

An automatic summary is a text generated by software, which is coherent and contains a significant amount of relevant information from the source text. Its compression rate τ is less than a third of the length of the original document. Generating summaries demands that the summarizer (both human and algorithm) makes an effort to select, reformulate and create a coherent text containing the most informative segments of a document. The notion of segments of information is left purposefully vague [9].

A well-prepared abstract enables readers to identify the basic content of a document quickly and accurately, to determine its relevance to their interests, and thus to decide whether they need to read the document in its entirety.

Why summarizing texts? There are several valid reasons in favor of the automatic summarization of documents. Here are just a few [21, 24]:

- ✓ Summaries promote current awareness.
- ✓ Summaries facilitate literature searches.
- ✓ Summaries aid in the preparation of reviews.
- ✓ Summaries reduce reading time.
- ✓ When researching documents, summaries make the selection process easier.
- ✓ Automatic summarization improves the effectiveness of indexing.
- ✓ Automatic summarization algorithms are less biased than human summarizers.
- ✓ Personalized summaries are useful in question-answering systems as they provide personalized information.
- ✓ Using automatic or semi-automatic summarization systems enable commercial abstract services to increase the number of texts they can process.

A process of constructing abstracts from the texts of the document has gained increasing popularity in recent years. Abstracts serve as quick guides to grasp the main subject of the document without reading the whole document. They help to decide the relevance of the document to the user. Text summarization is the useful task that takes advantage of the natural language processing techniques and helps in document searching and browsing in a variety of contexts such as: over the Internet, digital libraries and hand-held devices. Text summarization is also used for blogs as a preprocessing step for blog mining [21].

2.2 Types of Text Summarization

Summaries can be categorized according to different sets of criteria, such as their function, type and document source, etc. According to their function, Text Summarization can be indicative or informative [25]. Indicative summary provides information about the topics discussed in the source document. It resembles a table of contents. Informative summary aims to reflect the content of the source text, possibly explaining the arguments. It is a shortened version of the document. According to the number of documents for summarization it can be single document or Multi-document summary [26]. Single document is a summary of one document; Multi-document is a summary of not necessarily heterogeneous group of documents often about a specific topic. According to their type Text Summarization can be

extraction or abstraction. Extract is an assembly of fragments from the original document while abstract is summarizing by reformulating. Abstractive is a summary produced by rewriting and/or paraphrasing the text. Text summarization can be generic summary or query-guided summary based on the context. Generic summary is a summary of a document which ignores users' information needs. Whereas, query-based is a summary guided by information needs or by users' queries. In this research our work falls in the category of generic extractive single document summarization.

2.3 Summarization Parameters

There are many parameters that are used in summarization. The most common parameters used are as follows [21]:

- Compression rate: It is the amount of information the user needs from the source and it is usually set from 5% to 30 % depending on the application in which summarization is used. It is given by the ratio of the length of the summary to the length of the source document.
- Function: This parameter is used to select the type of summaries the user needs based on the content. It can be either indicative or informative.
- Audience: It is the parameter used to set the type of users of the summary. It can be user-focused or generic.
- Span: This parameter sets the language in which the summary is generated. Summaries can be monolingual or multilingual.
- Relation to the source: This parameter is used to select whether the user needs extractive or abstractive summary.
- Genre: This parameter is used to select the domain in which the summary is to be generated. It can be technical articles, sports articles, editorials, scientific articles, email messages, books and others.
- Media: This parameter is set to indicate the different forms of the summary such as text, tables, diagrams, speech, video, movies and others.
- Coherence: It is a crucial concept in summarization in which different textual units gather to form an integrated whole. This parameter is used to set the level of tolerance for incoherence. Some applications accept summaries which are just fragmentary in the form of a list of words or phrases and some applications may contain passages of connected

text. Significance of parameters may vary according to the application of summarization. Relevant parameters may be used to satisfy the purpose for which summarization is used.

2.4 Approaches used for Text Summarization

Text summarization approaches can be categorized into two depending on the degree of linguistic processing employed in the process of generating summaries: Shallow-processing and deep-processing [21]. Shallow-processing primarily employs very shallow features such as counting word frequencies, TF-IDF scores (i.e., term frequency multiplied by inverse document frequency), and sentence position. Shallow approaches use techniques which do not require the linguistic analysis beyond syntactic level. These methods tend to identify the salient portions of the text based on the surface level analysis of the document. They extract the sentences, considered salient, and then re-arrange them to form a coherent summary. The main advantage of these approaches is the robustness because it uses some straight forward methods to select summary sentences. However, there are some limitations in terms of the quality of the summary because it is hard to understand the real meaning of a sentence using these approaches. Since these methods extract the complete sentence(s), they cannot achieve greater compression rates compared to the deeper approaches. A classic example of shallow processing can be found in Edmundson [10].

Deep-processing performs deeper semantic analysis of the document content to identify the salient portions. They require highly domain-specific information to be able to perform deeper analysis. Since the output texts of such approaches are generated by the machine, it requires rich linguistic resources such as sentence parsers, morphological parsers, WordNet, domain specific corpora among others. These approaches were used for specific domains which have structured data as the input source such as the results and statistics of sport events, stock market bulletins and others. They produce more informative summaries since they are capable to identify more salient information of the input. Lack of such widely available knowledge bases factors makes these methods hard to implement. One major advantage of these methods is the level of compression obtained. Deep-processing employ more linguistic processes such as (partial) parsing, chunking, or semantic relationships.

2.5 Techniques used for Text Summarization

According to [19], the most important concepts useful to create a summarizer is to understand and decide appropriate technique to be used for creating it. To decide and identify the most important text units for the required summary, different researchers have been using one or a combination of different extraction features and weighting techniques to determine the summary to be produced.

Every extractive text summarization system consists of four basic steps. These steps are represented as a pipeline of ATS [27]. These pipelines are shown in Figure 2.1 [27].

Figure 2. 1: General overview of the elements of an extractive summarization method

2.5.1 Preprocessing

Working with unstructured text data is hard especially when we are trying to build an intelligent system which interprets and understands free flowing natural language just like humans. We need to be able to process and transform noisy, unstructured textual data into some structured, vectorized formats which can be understood by any machine learning algorithm [28].

Therefore, applying preprocessing to clean the document is almost obligatory [23]. Since, our work is text summarization which is one of NLP applications we have to conduct preprocessing as well. There can be multiple ways of cleaning and preprocessing textual data. The most important ones which are used heavily in NLP are described as follows.

I. Cleaning

In this module if the text has unnecessary content like HTML tags, which do not add much value when analyzing text, rare punctuation (especially all different quotes) will be removed.

II. Sentence boundary detection

Sentence boundary detection is the problem of deciding where sentences end. There is a straightforward solution in many cases (i.e., in most cases, splitting after a period or question/exclamation mark is the right decision).

III. Tokenization

Word tokenization is the problem of splitting a sentence into separate word tokens. While splitting on whitespace is an excellent heuristic, this approach fails in many cases. Quotation marks should be interpreted as separate tokens, for example, even though they are not whitespace-separated from the words they mark. In our research, we use the NLTK tokenizer by adding some rules to cover Afaan Oromo rules.

IV. Stemming

Stemmer in NLP and IR is an attempt to reduce a word to a common root or stem form [8]. Word stems are usually the base form of possible words that can be created by attaching affixes like prefixes and suffixes to the stem to create new words. This is known as inflection. The reverse process of obtaining the base form of a word is known as stemming [29].

V. Stop word removal

Words which have little or no significance especially when constructing meaningful features from text are known as stop words. These are usually words that end up having the maximum frequency if we do a simple term or word frequency in a corpus [28, 30]. In Afaan Oromo the words hin, jira, irra, bira and so on are considered to be stop words. So, stop word removal is an important preprocessing technique used in Natural Language processing applications to improve the performance of the Information Retrieval System, Text Analytics & Processing System, Text Summarization, Question-Answering system, stemming etc. In our research we used NLTK stop word remover. We gathered more than 200 stop words from different sources. We used stop word lists gathered by [8, 19].

2.5.2 Text Representation

For any simplest summarizer, intermediate representation of the text to be summarized is done to identify the important content. It is an essential step [31]. The aim of text representation for extractive summarization is to construct a similarity measure between sentences in the document [27]. Many sentence selection methods rely on similarities between all sentences in a text.

i. Sentence Similarity

To measure sentence similarity many authors used word or phrase overlap. Sentences which has more similar words/phrases is considered to be similar to each other. Document similarity

is the process of using a distance or similarity-based metric that can be used to identify how similar a text document is with any other document(s) based on features extracted from the documents like bag of words, bag of N-grams model or TF-IDF.

Bag of words

This is the simplest vector space representational model for unstructured text. A vector space model is simply a mathematical model to represent unstructured text (or any other data) as numeric vectors, such that each dimension of the vector is a specific feature/attribute. The bag of words model represents each text document as a numeric vector where each dimension is a specific word from the corpus and the value could be its frequency in the document occurrence denoted by 1 or 0 or even weighted values. The model's name is such because each document is represented literally as a 'bag' of its own words, disregarding word orders, sequences and grammar [29]. The model is only concerned with whether known words occur in the document, not where in the document [32].

Bag of N-Grams

As we discussed above bag of words doesn't consider order of words. But what if we also wanted to take into account phrases or collection of words which occur in a sequence? N-grams help us achieve that. An N-gram is basically a collection of word tokens from a text document such that these tokens are contiguous and occur in a sequence. Bi-grams indicate n-grams of order 2 (two words), Tri-grams indicate n-grams of order 3 (three words), and so on.

TF-IDF Model

There are some potential problems which might arise with the bag of words model when it is used on large corpora. Since the feature vectors are based on absolute term frequencies, there might be some terms which occur frequently across all documents and these may tend to overshadow other terms in the feature set [33]. TF-IDF is a simple, but powerful representation of sentences, indicating the relative importance of words in a sentence. Cosine similarity between TF-IDF vectors is still a popular baseline measure for its simplicity, efficiency, and effectiveness. The TF-IDF model tries to combat this issue by using a scaling or normalizing factor in its computation. TF-IDF stands for Term Frequency-Inverse Document Frequency, which uses a combination of two metrics in its computation, namely: term frequency (tf) and inverse document frequency (idf). This technique was developed for

ranking results for queries in search engines and now it is an indispensable model in the world of information retrieval and NLP.

Mathematically, TF-IDF can be defined as [15]:

$$\boldsymbol{TF - IDF = TF * IDF} \tag{1}$$

This can be expanded further to be represented as follows:

$$\boldsymbol{TF - IDF(w, D) = TF(w, D) * IDF(w, D) = TF(w, D) \log \frac{C}{IDF(W)}} \tag{2}$$

Here, TF-IDF(w, D) is the TF-IDF score for word w in document D. The term TF(w, D) represents the term frequency of the word w in document D, which can be obtained from the Bag of Words model. The term IDF(w, D) is the inverse document frequency for the word w, which can be computed as the log transform of the total number of documents in the corpus C divided by the document frequency of the word w, which is basically the frequency of documents in the corpus where the word w occurs. There are multiple variants of this model but they all end up giving quite similar results [29].

After getting feature vectors using the above methods there are many different similarity measures which can be used to measure the similarity. These include cosine distance/similarity, euclidean distance, manhattan distance, BM25 similarity, jaccard distance and so on. Since we used cosine similarity in our work, we only focus on cosine similarity. Cosine similarity basically gives us a metric representing the cosine of the angle between the feature vector representations of two text documents. The lower the angle between the documents, the closer and more similar they are [29].

According to [27], currently, the enormous number of sentence similarity measures could generally be divided into two groups. The first group comprises measures which are designed only with the task of similarity measurement in mind. Traditional (count-based) feature engineering strategies for textual data involve models belonging to a family of models popularly known as the Bag of Words model. This includes term frequencies, TF-IDF (term frequency-inverse document frequency), N-grams and so on. While they are effective methods for extracting features from text, due to the inherent nature of the model being just a bag of unstructured words, we lose additional information like the semantics, structure, sequence and

context around nearby words in each text document. Models which can capture this information and give us features which are vector representation of words are popularly known as embeddings.

The second group of methods do not solely aim at the task of measuring similarity, but on a more robust semantic representation of sentences intrinsically. These semantic representations are usually based on word-level semantic representations. Word embeddings are a good example for this group.

ii. Word Embedding

To overcome the shortcomings of losing out semantics and feature sparsity in bag of words model based features, we need to make use of vector space models (VSMs) in such a way that we can embed word vectors in this continuous vector space based on semantic and contextual similarity. The distributional hypothesis in the field of distributional semantics tells us that words which occur and are used in the same context are semantically similar to one another and have similar meanings. In simple terms, 'a word is characterized by the company it keeps' [34]. There are two main types of methods for contextual word vectors. Count-based methods like Latent Semantic Analysis (LSA) which can be used to compute some statistical measures of how often words occur with their neighboring words in a corpus and then building out dense word vectors for each word from these measures. Predictive methods like Neural Network based language models try to predict a word from its neighboring words looking at word sequences in the corpus and in the process, it learns distributed representations giving us dense word embeddings. Since we use Word2Vec, we will be focusing on predictive methods in this research.

In every natural language, sentences are constructed from words. The idea that similar words occur in similar context has become the dominant approach in semantic representation: distributed representations of words have become the foundation of many modern solutions for NLP problems. The simple idea of representing a word as a prediction of its context, thereby yielding a continuous representation of a word embedded in a vector space, turns out to be a very robust representation of a word's meaning [27].

Various unsupervised methods towards continuous word embeddings are available. The most popular and well-explored method is Word2Vec [35]. It naturally captures linguistic relations such as (dis)similarity and word analogies, both in the semantic and syntactic sense [36]. This

model was created by Google in 2013 and is a predictive deep learning-based model to compute and generate high quality, distributed and continuous dense vector representations of words, which capture contextual and semantic similarity. Essentially these are unsupervised models which can take in massive textual corpora, create a vocabulary of possible words and generate dense word embeddings for each word in the vector space representing that vocabulary. Usually we can specify the size of the word embedding vectors and the total numbers of vectors are essentially the size of the vocabulary. This makes the dimensionality of this dense vector space much lower than the high-dimensional sparse vector space built using traditional bag of words models.

There are two different model architectures which can be leveraged by Word2Vec to create these word embedding representations. These include,

- ✓ The continuous bag of words (CBOW) Model
- ✓ The skip-gram model

The Continuous Bag of Words (CBOW) Model

The CBOW model architecture tries to predict the current target word (the center word) based on the source context words (surrounding words). Considering a simple sentence, "***the quick brown fox jumps over the lazy dog***", this can be pairs of (context_window, target_word) where if we consider a context window of size 2, we have examples like ([quick, fox], brown), ([the, brown], quick), ([the, dog], lazy) and so on. Thus, the model tries to predict the

target_word based on the context_window words. Both CBOW and skip-gram are represented in Figure 2.2 [36].

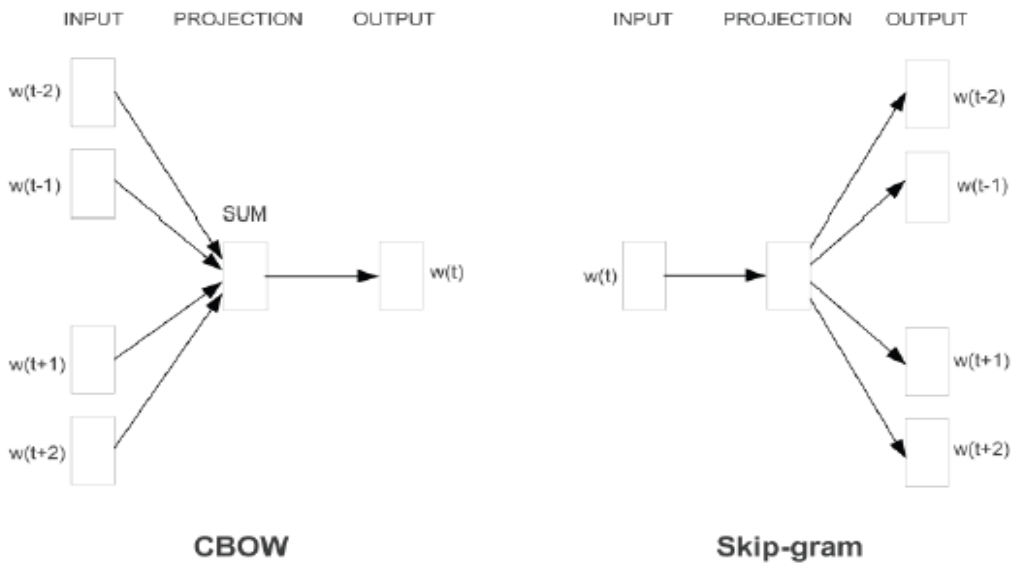

Figure 2. 2: Model of CBOW and Skip-gram

The context is defined as the 'window' of words around a target word, where typical window sizes are between 2 and 5 words. This is illustrated in Figure 2.3 [36]. The window around the target word defines the set of context words used for training the skip-gram model

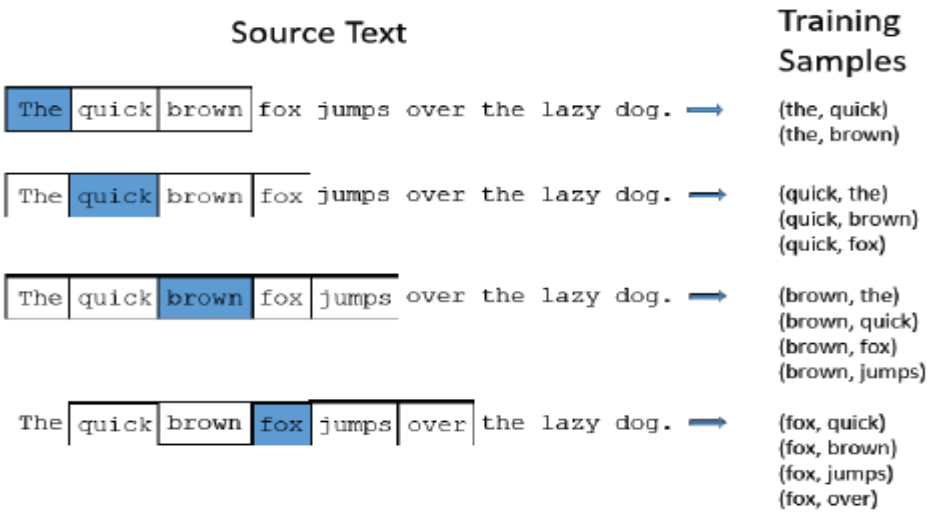

Figure 2. 3: Example of the target and context words for a sentence, with a window

The Skip-gram Model

The Skip-gram model architecture usually tries to achieve the reverse of what the CBOW model does. It tries to predict the source context words (surrounding words) given a target word (the center word). Considering our example, we used earlier, "***the quick brown fox jumps over the lazy dog***". The skip-gram model's aim is to predict the context from the target word; the model typically inverts the contexts and targets, and tries to predict each context word from its target word. Hence, the task becomes to predict the context [quick, fox] given target word 'brown' or [the, brown] given target word 'quick' and so on. Thus, the model tries to predict the context window words based on the target word. Skip-gram has been represented in Figure 2.3 [37].

Word2Vec is always trained on a large corpus of tokenized sentences. For skip-gram, the general steps are as follows. First, a vocabulary will be built by passing over the corpus once, keeping all words that occur more than n times. Then two random vectors of arbitrary dimensionality will be initialized (typically 300) for each word in the vocabulary: a 'target' vector representing its meaning, and a 'context' vector to represent the separate 'context meaning' of a word. Combining all vectors in a matrix, thus two |V| ⇥ D matrices will be obtained, where D is the number of dimensions, and |V| is the number of words in the vocabulary. These matrices will be named the embedding matrix, containing the actual embeddings of each word, and the context matrix, containing the 'context embeddings' of each word. The goal is to train the initial random D-dimensional target vectors to predict their context, thereby representing the target word's meaning. For every training iteration, a target word wt from the corpus will be sampled, estimate the probabilities of the context words wc given the target word wt [35].

2.5.3 Sentence Selection

The general objective for sentence selection is to select maximally informative sentences, without information overlap between the sentences, to maximize the coverage of the original article by the summary [27]. Although the number of sentence selection methods is too large to provide a full overview, some well-known methods which are commonly used as baselines are discussed below.

a. Maximal marginal relevance

A well-known, classic, unsupervised algorithm used for sentence selection is the Maximal Marginal Relevance (MMR) algorithm. MMR is an algorithm that balances the trade-off between relevance or informativeness and coverage, thereby incorporating diversity into the summary. Using a greedy approach[1] selecting one sentence every iteration, it picks the sentence from the article that is both relevant and dissimilar from already picked sentences until a word maximum is reached. It is an efficient method, as it only depends on the already picked sentences and the candidate sentence's own informativeness in every iteration [27].

b. Centroid-based: MEAD

Another selection method is a centroid based algorithm MEAD [38]. It assumes a semantic vector representation of every sentence. It then clusters the sentences and computes centroids for each cluster. Combining centroid-sentence cosine similarities and other sentence features (i.e. length) to score all sentences in each cluster, a subset of sentences of each cluster is then selected to form a summary [27].

c. Topic-based

There are also summarizers which first detect the most important topics of a document, and then extract sentences to maximize the coverage of these topics. In contrast to the topic signatures, which represent the topic of a document using a group of words, these methods represent a document using a set of pre-defined topics. Each topic corresponds to a group of correlated words or distribution in which some correlated words are most frequent. Because of this, topic modeling can abstract away the variations of surface expression to a certain extent, for example, by regarding different terms as expressing the same topic [39].

In [40], the authors proposed an approach which combines the automatic topic identification with term frequency methods. This methodology consists of calculating initially the similarity between the sentences and then carries out the identification of the subject covered by gathering similar sentences in clusters. In a second stage based on terms frequency, the projecting sentences are selected starting from the local topics already identified.

[1] A greedy algorithm is any algorithm that follows the problem-solving heuristic of making the locally optimal choice at each stage with the intent of finding a global optimum.

d. Discourse

Discourse approach is used in linguistic techniques for ATS. They exploit the discursive organization of a text to improve the relevance and quality of summaries. The discursive organization of a text implies the global structure of the text and it includes format of a document, threads of topics in the text and rhetorical structure of the text. This approach asserts that texts are not just a linear sequence of clauses and sentences but they are a cluster of clauses and sentences, called discourse segments that are related pragmatically to form a hierarchical structure. Thus, in this approach, a text is first divided into discourse segments and based on these segments, the discourse structure of the text as intended by its author is re-constructed. This discourse structure or discursive representation of the text has been shown to be one way of determining the most important units of a text [20]. The relation or connection between sentences and parts in the text are represented by discourse relations.

e. Graph-based

A popular class of sentence selection methods is graph-based methods. Graph-based methods are usually based on Google's famous PageRank algorithm and are generally considered more flexible; while centroid-based methods define a hard clustering on all sentences, the graph-based approach allows sentences to be more or less connected to each other [27].

This method consists in building a graph from the text. The graph is capable of representing different phenomena where relations between objects are important, and one of such phenomena is text summarization. In text summarization, a given document D is represented as graph G = (V, E), where the graph G= (V, E) is an undirected weighted graph that represents document D with a set of vertices V and edge between vertices E [17]. In a graph representation, the content overlap between sentences is reflected by edges, and the importance of a sentence is determined by its relations with other sentences. In other words, the graph of sentences is a very basic model of the content structure of a document. There are two main algorithms used in graph-based approach: PageRank and HITS (Hypertext Induced Topic Search). Google has the most well-known ranking algorithm called PageRank that has been claimed to supply top ranking pages that are relevant. The PageRank was used and enhanced by Page and Brin [41]. PageRank [41] is an algorithm for computing eigenvector centrality based on the concept of a random walk. Consider a surfer who begins at a node and randomly proceeds to another node following an edge, the importance of a node is reflected

by the probability of the surfer visiting that node. HITS involve computation of two scores for each node in the graph: hub and authority score. The hub score is a measure of the outgoing links of a node whereas the authority score measures the scores of incoming links of a node. These algorithms originally assumed unweighted directed graphs [17].

f. Latent Semantic Analysis (LSA) Based Approaches

LSA is an algebraic-statistical method that extracts and represents semantic knowledge of the text based on the observation of the co-occurrence of words. This technique aims to build a semantic space with very large dimension from the statistical analysis of the whole co-occurrences in a corpus of texts. The starting point of LSA consists of a lexical table which contains the number of occurrences of each word in each document [21].

In LSA, the meaning of sentences and the meaning of words can be represented simultaneously. The meaning of the sentence can be found by averaging the meaning of words that the sentence contains, and the meaning of words are represented by averaging the meaning of sentences that contain this word. LSA method uses Singular Value Decomposition (SVD) for finding out semantically similar words and sentences. SVD is a method that models relationships among words and sentences. It has the capability of noise reduction, which leads to an improvement in accuracy [8].

LSA has three main limitations. The first limitation is that it uses only the information in the input text, and it does not use the information of world knowledge. The second limitation is that it does not use the information of word order, syntactic relations, or morphologies. Such information is used for finding out the meaning of words and texts. The third limitation is that the performance of the algorithm decreases with large and heterogeneous data. The decrease in performance is observed since SVD, which is a very complex algorithm, used for finding out the similarities.

g. Machine learning approaches

Machine learning can be thought of as a way to combine different features being proposed for sentence extraction [21]. The basic idea of the machine learning approach is: given a training set of documents with manually selected extractive summary, develop a classification function that estimates the probability of a given sentence to be included in an extract. Machine learning approaches have been proved very successful in domain-specific summarization where classifiers can be trained to identify specific type of information. In scientific articles,

sentences describing conclusion part are important or for minutes of meetings, utterances expressing agreement or disagreement are important. The two common machine learning approaches are Naive-Bayes and Hidden Markov Model. Another method in this approach involves training the neural networks to learn the types of sentences that should be included in the summary. This is accomplished by training the network with sentences in several test paragraphs where each sentence is identified as to whether it should be included in the summary or not. This is done by a human reader. The neural network learns the patterns inherent in sentences that should be included in the summary and those that should not be included. It uses three-layered Feed forward neural network.

h. Supervised Sentence Classifiers: Recurrent Neural Networks

Currently Supervised approaches towards extractive and abstractive summarization are emerging. One of such supervised approaches is a recurrent neural network (RNN). RNN remembers the past and its decisions are influenced by what it has learnt from the past. It uses Encoder-Decoder architecture as a way of organizing recurrent neural networks for sequence prediction problems that have a variable number of inputs, outputs, or both inputs and outputs. Both the encoder and the decoder sub-models are trained jointly [42]. A recent, very successful approach was proposed in [43]. The authors propose a deep learning approach by utilizing recurrent neural network (RNN) architecture to 'read' all sentences of a document, combined with a single-layer convolutional neural network (CNN) to learn to encode words into sentences. The authors propose both an abstractive and an extractive approach. They essentially treat extractive summarization as a sentence labeling task using RNN architecture to both encode the document and label the salience of every sentence. Sentence importance is learned from a gold standard of hundreds of thousands of sentence-labeled news articles, based on pairs of news articles and bullet-point summaries from CNN and DailyMail. This approach yields good results on the dataset itself but also appears to generalize well to the DUC-2002 dataset.

In this thesis we have used word embedding with graph-based approach. Graph based approaches capture redundant information and they improve coherency by capturing relationships between two sentences.

2.6 Evaluation of Text summarization

Evaluating a summary is also a most important task. According to [44] summary evaluation methods are divided into two. Extrinsic evaluation and intrinsic evaluation. The general taxonomy is shown in Figure 2.4.

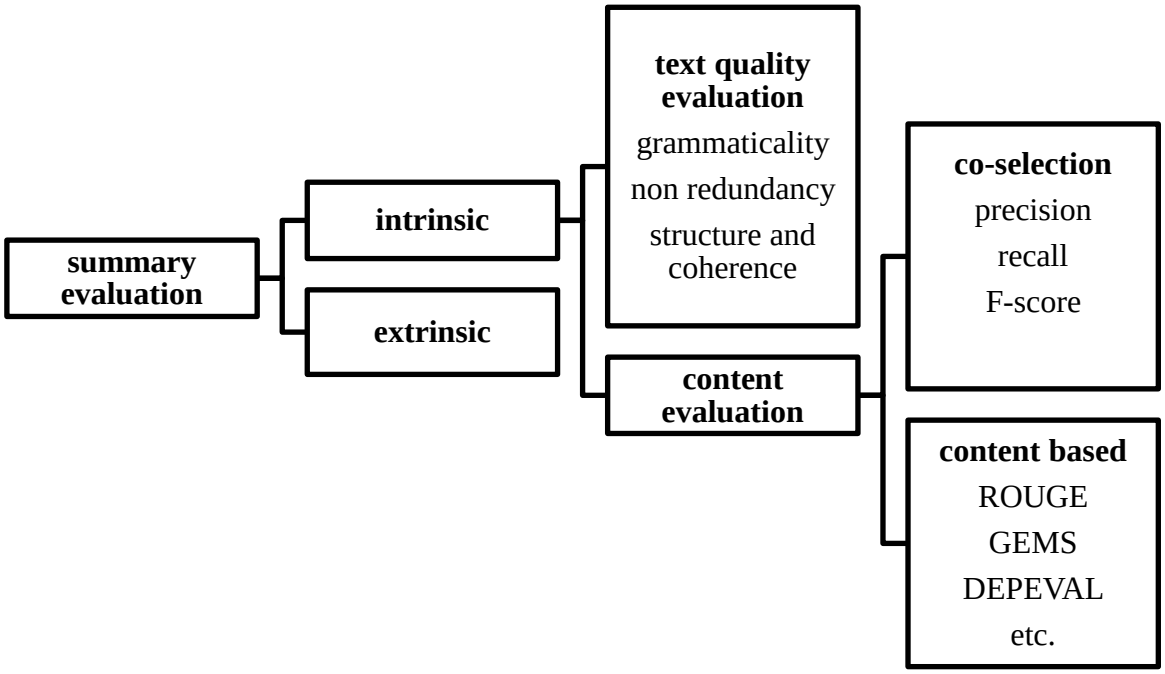

Figure 2. 4: The Taxonomy of Summary Evaluation Measures

Extrinsic Evaluation

It checks the summarization based on how it influences the completion of some other tasks such as text classification, information retrieval, question answering, etc. It evaluates the impact of summarization on tasks like reading comprehension, relevance assessment, etc. Therefore, a summary is considered as good if it is helpful to some other tasks [44].

1. Reading comprehension: This method determines whether it is capable to answer multiple-choice tests after comprehension of the summary.
2. Relevance assessment: A variety of methods are used for evaluating the relevance of the subject present in the summary of the original document.

Intrinsic Evaluation

It checks the summarization system itself. It determines the summary quality based on a comparison between the automatically generated summary and the human-made manual

summary. A summary is evaluated based on two aspects quality or informativeness. The informativeness of a summary is evaluated by comparing it with a human-made summary, i.e., reference/ ideal summary. There is another model called fidelity to the source which checks whether the summary consists of the same or similar content as present in the original document.

Informativeness evaluation

There are plenty of methods used for informativeness evaluation. Some of them are ROUGE (recall-oriented understudy of gisting evaluation), GEMS (Generative Modelling for Evaluation of Summaries), DEPEVAL (dependency evaluation), ParaEval (paraphrase evaluation), AutoSummENG (Automatic Summary Evaluation based on N-gram Graphs) and so on.

Other popular metrics: For intrinsically evaluating the summary are precision, recall, and F-measure [44]. They are required to predict coverage between human-made ideal summary and automatically generated machine-made summaries.

These metrics are explained below:

i. Precision: It determines what fraction of the sentences chosen by a human and selected by the system are correct. Precision is the number of sentences found in both systems and ideal summaries divided by the number of sentences in the system summary.
ii. Recall: It determines what proportion of the sentences chosen by humans is even recognized by the machine. A recall is the number of sentences found in both systems and ideal summaries divided by the number of sentences in the ideal summary.
iii. F-measure: It is computed by combining recall and precision.

They are calculated as follows:

- ✓ Recall (R) = TP / (TP + FP)
- ✓ Precision (P) = TP/ (TP + FN)
- ✓ F-measure (F) = (2*R*P) / (R+P)

 Where:

- ✓ TP= true positive FP= False Positive, TN = True negative, FN= false negative
- ✓ TP: Manually generated intersection with machine generated summary
- ✓ TP + FP: Machine generated summary
- ✓ TP + FN: Manually generated summary

Quality evaluation

In Quality evaluation, linguistic aspects of the summary are considered. In the conferences of DUC and TAC, the following factors related to linguistic quality are used for evaluating summaries.

a) Redundancy: The summary should not contain redundant information.
b) Grammaticality: The text should not contain non-textual items (i.e., markers) or punctuation errors or incorrect words.
c) Referential clarity: The nouns and pronouns should be referred to in the summary. For example, the pronoun 'she' has to mean that it is referring to somebody in the context of the summary.
d) Structure and Coherence: The summary should have good structure and the sentences should be coherent.

These do not need to be compared against the ideal summary. Expert human evaluator evaluates the summary manually by assigning a score to the summary corresponding to a five-point scale based on its quality.

Chapter Three: Afaan Oromo

In this chapter, Afaan Oromo Alphabet and writing system, punctuation marks and usage, Afaan Oromo morphology, Afaan Oromo word, and sentence boundaries are discussed.

3.1 Afaan Oromo Alphabets

Afaan Oromo is a phonetic language, which means that it is spoken in the way it is written. The writing system of the language is straightforward which is designed based on the Latin script. Unlike English or other Latin-based languages, there are no skipped or unpronounced sounds/alphabets in the language. Every alphabet is to be pronounced in a clear short/quick or long/stretched sounds. In a word where consonant is doubled the sounds are more emphasized. Besides, in a word where the vowels are doubled the sounds are stretched or elongated [19].

Afaan Oromo has the same vowels and consonants as English. Afaan Oromo vowels are represented by the five basic letters such as **a, e, i, o, u**. Consonants, on the other hand, do not differ greatly from English, but there are few special combinations such as "**ch**" and "**sh**" (same sound as English), "**dh**" in Afaan Oromo is like an English "**d**" produced with the tongue curled back slightly and with the air drawn in so that a glottal stop is heard before the following vowel begins. Another Afaan Oromo consonant is "**ph**" made when with a smack of the lips toward the outside "**ny**" closely resembles the English sound of "**gn**". We commonly use these few special combination letters to form words. For instance, **ch** used in **kofalchiisaa** 'making laugh', **sh** used in **shamarree** 'girl', **dh** used in **dhadhaa** 'butter', **ph** used in **buuphaa** 'egg', and **ny** used in **nyaata** 'food'. In general, Afaan Oromo has 31 letters (26 consonants including special combinations and 5 vowels) called "**Qubee**" [45]. All the alphabets of Afaan Oromo are presented in Figure 3.1.

In general, all letters in the English language are also in Afaan Oromo except the way it is written.

A a	B b	C c	CH ch		D d	DH dh		E e	F f	G g	H h	I i
J j	K k	L l	M m	N n	NY ny		O o	P p	PH ph		Q q	R r
S s	T t	U u	V v	W w	X x	Y y	Z z					

Figure 3. 1: Afaan Oromo Alphabets/ Qubee Afaan Oromo

Words

The word is the smallest unit of a language. There are different methods for separating words from each other. However, most of the world languages including English use the blank character (space) to show the end of a word. Some long words are being cut in written form (abbreviation), with the symbols "/", ".", and therefore this symbol should not determine a word boundary. The usual parenthesis, brackets, quotes, all kinds of marks, are being used to show a word boundary in Afaan Oromo [46].

Sentence

Afaan Oromo sentence is terminated like English and other languages that follow the Latin writing system [46]. That means, the full stop (.) in a statement, the question mark (?) in interrogative and the exclamation mark (!) in command and exclamatory sentences, mark the end of a sentence and the comma (,) which separates listing in a sentence and the semicolon is to mark a break that is stronger than a comma but not as final as a full stop balance.

3.2 Afaan Oromo Morphology

Morphology is a branch of linguistics that studies and describes how words are formed in a language [47]. There are two types of morphology: inflectional and derivational. Inflectional morphology is concerned with the inflectional changes in words where word stems are combined with grammatical markers for things like a person, gender, number, tense, case, and mode. Inflectional changes do not result in changes in parts of speech. On the other hand, derivational morphology deals with those changes that result in changing classes of words

(changes in the part of speech). For instance, a noun or an adjective may be derived from a verb.

Types of Morphemes in Afaan Oromo

A morpheme is the smallest semantically meaningful unit in a language. A morpheme is not identical to a word, and the principal difference between the two is that a morpheme may or may not stand alone, whereas a word, by definition, is a freestanding unit of meaning. Every word comprises one or more morphemes. In Afaan Oromo, there are two categories of morphemes: free and bound morphemes. A free morpheme can stand as word on its own whereas bound morpheme does not occur as a word on its own [45, 46]. In Afaan Oromo roots (stems) are bound as they cannot occur on their own. Example: "**dhug-**" (drink) and "**beek-**" (know), which are pronounceable only when other completing affixes are added to them [19].

Similarly, an affix is also a bounded morpheme that cannot occur independently. It is attached in some manner to the root, which serves as a base. These affixes are of three types – prefix, suffix, and infix. The first and the second types of affixes occur at the beginning and at the end of a root respectively in creating a word whereas the infix occurs in between characters of the word. In **dhugaatii** 'drink', for instance, **-aatii** is a suffix and **dhug-** is a stem. Moreover, an infix is a morpheme that is inserted within morpheme. In the work of [47] it is discovered that Afaan Oromo does not have infixes like English.

There are many ways of word formation in Afaan Oromo. These morphological analyses of the language are organized into six categories [47]. The categories are nouns, verbs, adjectives, adverbs, functional words, and conjunctions. Almost all Afaan Oromo nouns in a given text have person, number, gender, and possession markers which are concatenated and affixed to a stem or singular noun form. Afaan Oromo verbs are also highly inflected for gender, person, number, and tenses. Adjectives in Afaan Oromo are also inflected for gender and number. Moreover, adverbs can be categorized into adverbs of time, adverb of place, and adverb of how some of the adverbs are affixed.

Furthermore, functional words can be classified as prepositions; postpositions, and articles markers which are often indicated through affixes in Afaan Oromo. Lastly, conjunctions can be separate words (subordinating or coordinating), and some of them are affixed. Since Afaan Oromo is morphologically very productive, derivation, reduplication, and compounding are

also common in the language. The following is detail descriptions and examples of the word-formation process of Afaan Oromo based on the works of [45, 48].

3.2.1 Afaan Oromo Nouns

I. **Gender**

Gender is one category of nouns, pronouns, and adjectives into masculine and feminine and some language neuter based on whether a noun is considered as male, female, or without sex respectively.

Gender is of two types: natural and grammatical.

Natural gender refers to the natural sex of animate things.

Example

Abbaa father -**Haadha** mother

Dhirsa husband -**Niitii** wife

Most nouns are not marked by gender affixes. Only a limited group of nouns differ by using different suffixes for the masculine and the feminine form. Grammatically the language uses **-ssa** for masculine and **-ttii** for the feminine [48].

Obboleessa brother - **obboleettii** sister

Ogeessa expert (male) - **Ogeettii** expert (female)

Natural female gender corresponds to the grammatical feminine gender. The sun, moon, stars, and other astronomic bodies are usually feminine. In some Afaan Oromo dialects, geographical terms such as names of towns, countries, rivers, etc. are feminine, in other dialects such terms are treated as masculine nouns. It is due to this fact that there are different subject forms for the noun **biyya** 'country' [46].

Example: **biyyi**(male) or **biitti** (female).

There are also suffixes like **-a**, **-e** that indicate a present and past form of masculine markers respectively. **-ti** and **-tii** for present feminine marker and **-te** past tense marker, **-du** for making adjective form [47].

Biiftuun **baate** 'the sun rose'.

The word **baate** takes **-te** to show feminine gender. We can see that **-tii** can also show feminine gender in the following statement.

Adurreen maal **ariitii**? What does the cat run after?

II. Number

Afaan Oromo has different suffixes to form the plural of a noun. The use of different suffixes differs from dialect to dialect. In connection with numbers the plural suffix is very often considered unnecessary: **harka ishee lamaaniin** with her two hand(s).

The majority of plural nouns are formed by using the suffixes [19, 48].

– **oota**, followed by –**lee**, **-wwan** , **-een**, **-olii**/ **-olee** and –**aan**. Other suffixes like –**iin** in **sariin** (**dogs**) are found rarely. Table 3.1 shows examples of Afaan Oromo plural noun suffixes.

Table 3. 1: Afaan Oromo Plural Noun Suffixes

Singular noun	Transliteration	Plural	Transliteration
/barataa/	'student'	/barattoota/	'students'
/farda/	'horse '	/fardeen/	' horses’
/gangee/	'mule'	/gaangoolii/	'mules'
/kitaaba/	' book'	/kitaabolee/	'books'
/bineensa/	' animal'	/bineensolii/	'animals'

III. Definiteness

Definiteness is a grammatical category used for distinguishing noun phrases according to weather their reference in a given context is presumed to be uniquely identifiable. In Afaan Oromo demonstrative pronouns like "**kun**" (this), **sun** (that) is used to express definiteness.

Mucaan kun this/ the Child (Subject)

Mucaa kana this/ the Child (Object)

Mucaan sun that/ the Child (Subject)

Mucaa sana that / the Child (Object)

To express indefiniteness emphatically the Oromo speaker my use numerical **tokko** one,

Example: **Muka tokko** one / a Tree.

In some Afaan Oromo dialects the suffix -**icha** (male), -**ittii**(n)(female) which usually has a singularize function is used where other languages would use a definite article.

Example:

Jaarsicha the old man (Subject) **Jaartittii** the old man (Subject)

IV. Derived Noun Forms

The most common word formation methods in Afaan Oromo are derivational and compounding [19].

Derivation

Derivational suffixes are added to the root or stem of the word. From derived verbal stem and adjectives may be formed by means of derivational suffixes. The following suffixes play an important role in Afaan Oromo word derivation. They are **-eenya, -ina, -ummaa, -annoo, -ii, -ee, -a, -iinsa, -aa,-i(tii), -umsa, -oota, -aata,** and **–ooma.**

Examples:

jab<u>aa</u>	strong	**jab<u>eenya</u>**	strength
jab<u>ina</u>	strength	**jab<u>ee</u>**	intensive
jab<u>ummaa</u>	strength	**jab<u>aachuu</u>**	to be strong
jab<u>aachisuu</u>	to make strong	**jab<u>eessuu</u>**	to make strong
<u>ja</u>jab<u>aachuu</u>	to be consoled	**jab<u>eefachuu</u>**	to make strong for one self

V. Compound words

On the other hand, it seems that the use of genitive constructions is a very old method of forming compound nouns, as traditional titles shown.

abbaa gadaa traditional Oromo president

abbaa caffee chairman of the legislative assembly

abbaa dubbii chief speaker of the caffee assembly

abbaa duulaa traditional Oromo minister of war

3.2.2 Afaan Oromo Verbs

Verbs are content words that denote an action, occurrence, or state of existence. Afaan Oromo has base (stem) verbs and four derived verbs from the stem. Moreover, verbs in Afaan Oromo are inflected for gender, person, number and tenses.

i. Derived stems

The four derived stems the formation of which is still productive in Afaan Oromo are:

Autobenefactive (AS)

Passive (PS)

Causative (CS)

Intensive (IS)

Passive, causative, and autobenefactive are formed with addition of a suffix to the root, yielding the stem that the inflectional suffixes are added to. The personal terminations according to different conjunctions are added to these affixes. The intensive stem is formed by reduplicating the first consonant and vowel of the first syllable of the stem. The derived stems may be formed from all verbs the meaning of which permits it [19, 46].

a. Autobenefactive

The Afaan Oromo autobenefactive (or "middle" or "reflexive-middle") is formed by adding **(a)adh**, **-(a)ach or -(a)at** or sometimes **-edh, -ech** or **–et** to the verb root.

This stem has the function to express an action done for the benefit of the agent himself.

Example: **bitachuu** to buy for oneself the root verb in this case is bit-

The conjugation of a middle verb is irregular in the third person singular masculine of the present and past (**-dh** in the stem changes to **-t**) and in the singular imperative (the suffix is **-u** rather than–**i**).

Examples:

bit- buy **bitadh-** buy for oneself

Infinitive and participles are always formed with **-(a)ch,** while the imperative forms have **-(a)(a)dh** instead of **-(a)ch**.

Infinitive	Imperative singular.	Imperative plural.	English
argachuu	**argadhu**	**argadhaa**	to find/get

b. Passive

The Afaan Oromo passive corresponds closely to the English passive in function. It is formed by adding -am to the verb root. The resulting stem is conjugated regularly.

Example: **beek-** know **beekam-** be known

c. Causative

The Afaan Oromo causative of a verb corresponds to English expressions such as: cause, make, let. It is formed by adding **-s**, **-sis**, or **-siis** to the verb root example:

Deemuu to go **deemsisuu** to cause to go

d. Intensive

It is formed by duplication of the initial consonant and the following vowel, geminating the consonant.

Example: **Waamuu** to call, invite **wawwaamuu** to call intensively

ii. Simple tenses

a. Infinite Forms

Infinite forms can be formed in two ways: Infinitive and Participle/gerund

Infinitive

Infinitive is an uninflected form of the verb. In Afaan Oromo infinitive form of verbs terminates in **-uu**. Examples:

arguu to see **deemuu** to go

On the other hand, the infinitive forms of autobenefactive verbs terminate in **-chuu**.

Example: **jiraachuu** to live **bitachuu** to buy for oneself

Participle/ gerund

Participle is a non-finite form of the verb whereas a gerund is a noun formed from a verb (in English the '**-ing'** form of a verb when used as a noun). In Afaan Oromo a participle is formed by adding **-aa** to the verb stem [48].

Example:

deemaa going **jiraachaa** living

According to the meaning of the verb these forms may serve as agent nouns.

barsiisaa teacher **gaafatamaa** responsible person

For these agent nouns feminine forms are used according to the pattern of feminine adjective formation.

barsiiftuu teacher **gaafatamtuu** responsible person

On the other hand, a gerund is formed by adding **-naan** to the verb stem.

deemnaan after having gone **nyaannaan** after having eaten

b. **Imperative**

Imperative singular of base stems and all derived stems beside autobenefactive stems is formed by means of the suffix **-i**. Example:

deemi! go! **argi**! look!

The imperative singular of autobenefactive stems is formed by means of the suffix **-u**. Example:

jiraadhu! live!

Imperative plural of all stems is formed by means of -**aa**.

Example: **deemaa**! go! **argaa**! see!

Negative imperatives are formed by means of - (**i**)**in** for singular and -(**i**)**inaa** for plural.

Example: **Qubaan jechoota irra hin deemiin.** Don't point on the words with your finger.

c. Finite Forms

The Afaan Oromo uses different conjugations for the verbs in main clauses and in subordinated clauses for actions in present or near future. The first-person singular is differentiated from the third person masculine by means of an -n that normally is suffixed to the word preceding the verb.

1. Present Tense Main Clause Conjugation

The present tense main clause conjugation is characterized by the vowel -**a**:

deemuu	to go
1.p. S	**deema**
2.p.	**deemta**
3.p.m	**deema**
3.p.f	**deemti**
l.p. pl	**deemna**
2.p. and polite form	**deemtu/deemtan(i)**
3.p. and polite form	**deemu/deeman(i)**

Examples: **gara mana barnootaan deema**. I go to the school.

2. Past tense conjugation

The past tense conjugation is characterized by the vowel -**e**:

deemuu	to go
3.p.S	**deeme**
3.p	**deemte**

3.p.m	**deeme**
3.p.f	**deemte**
l.p.pl	**deemne**
2.p. and polite form	**deemtani**
3.p. and polite form	**deemani**

Example: **gammachiis gara mana yaalaa deeme.** gammachiis went to the school.

3. Subordinate Conjugation

The subordinate conjugation is used in affirmative subordinated clauses and in connection with the particle **akka** for the jussive. Beside this the subordinate conjugation is used to negate present tense actions.

Deemuu	to go
l.p.S	**akkan deemu**
2.p.	**akka deemtu**
3.p.m.	**akka deemu**
3.p.f.	**akka deemtu**
l.p.pl	**akka deemnu**
2.p. and polite form	**akka deemtani**
3.p.and polite form	**akka deemani**

Examples: **Akkan yaadutti barnooti jira**. As I thought there is a school.

4. Contemporary verb conjugation

The contemporary verb conjugation is used only in connection with the temporal conjunction **-odoo,-otoo,-osoo,-otuu** or **-utuu** that being connected with this conjugation means ‘while’. The contemporary verb conjugation is a kind of subordinated conjugation with lengthened final vowels.

Example: **"Otuun isin waamuu maaliif deemtu?" jedhe**. He said, "While I was calling you (pl.) Why do you go?".

5. Jussive

To form the jussive in Afaan Oromo the particle **haa** has to be used in connection with the subordinate conjugation.

Example: **Isaan haa deemani** they shall go

6. Negation

Present tense main clause actions are negated by means of the negative particle **hin** and the verb in subordinate conjugation.

Example: **Dastaan hin jiru**. Desta is not present.

Present tense actions in subordinated clauses are negated by means of the negative particle **hin** and a suffix **-ne** that is used for all persons. Past tense actions are negated in the same way using the particle **hin** and the suffix **-ne**.

Example: **Ani dhufuu hin danda'u**. I can't come

iii. Verb Derivation

Some Afaan Oromo verbs are derived from nouns or adjectives by means of an affix **-oom**. These verbs usually express the process of reaching the state or quality that is expressed by the corresponding noun or adjective. From these process verbs causative and autobenefactive stems may be formed.

Examples: **danuu** much, many, a lot **guraacha** black

danoomuu to become much **gurraachomuu** to become black

Causative verbs, however, can also be derived directly from adjectives or nouns by suffixing a causative affix **-eess** to the stem of the noun or adjective, example:

danuu much **daneessuu** to increase, multiply

Another means to derive process verbs from adjectives in Afaan Oromo is to form an autobenefactive stem.

Example: **Adii** white **addaachuu** to become white

iv. Compound Verbs

In addition to the above discussed derived verbs, compound verbs can be formed by means of pre-/postpositions, pronouns and adverbs in Afaan Oromo such **as ol** above, **gad** below, **wal**, **waliin, walitti, wajjin** together**, keessa in, jala** under; they precede different verbs and express a broad variety of meanings [47].

Examples: **gadi dhiisuu** to let go of **gaddhiisuu** to let go of

Compound verbs can also be formed with **jechuu** or **gochuu**.

Example: With **jechuu** with **gochuu**

cal jechuu (to be quiet, silent) **cal gochuu** (to make quiet silent)

v. 'To be' and 'to have'

Afaan Oromo has different means to express 'to be'. One of them is copulas, other means are the verbs **ta'uu, jiruu** and **turuu.**

The morphemes **(-)dha** and **(-)ti** (suffixed or used as independent words) serve as affirmative copulas as well as the vowel -**i** that is added to nouns terminating in a consonant. The copula **dha** is used only after nouns terminating in a long vowel.

Negative copula is **miti**, irrespective of the termination of the noun.

Examples:

Present tense: **Anis jabaa <u>dha</u>**. I am strong, too.

Nouns terminating in a short vowel do not take any copula.

Example: **Isheen durba**. She is a girl.

Nouns and pronouns terminating in a consonant are combined with the copula.

Example: **Kuni bisbaani.** This is water.

In all utterances related to possession only the copula -**ti** may be used.

Example: **Hojiin hundee guddinaa ti**! Work is the basis of development.

Present progressive:

Waa'een jarreen Axaballaa warra isaaniitiif qofa otuu hin taane uummata naannoofiyyuu hibboo ta'aa iira.

The life of Axaballaa is like a mystery not only, for his family, but also for the people around him.

vi. Past Tense

Sangaan kan eenvuu ture? Whose ox was it?

The forms of the verb **qabuu** 'to have' are overlapping with the forms of the verb **qabuu** 'to grasp', 'keep'.

The verb **qabuu** appears with the meaning 'to have' only in the present tense and one past tense form. In present tense conjugation both verbs have the same form.

3.2.3 Afaan Oromo Adjectives

An adjective is a word which describes or modifies a noun or pronoun. A modifier is a word that limits, changes, or alters the meaning of another word. Unlike English adjectives are usually placed after the noun in Afaan Oromo. For instance, in **Tolaan farda adii bite** "Tola bought white horse" the adjective **adii** comes after the noun **farda**. In Afaan Oromo sometimes it is difficult to differentiate adjective from noun [46].

Example:

dhugaa truth, reality, true, right

dhugaa keeti your truth/ you are right (truth served as noun)

obboleessi hiriyaa dhugaati brother is the friend for truth / brother is a true friend (true served as adjective)

I. Gender

In Afaan Oromo adjectives are inflected for gender. We can divide adjectives into four groups with respect to gender marking. These are:

a. In the first group the masculine form terminates in –**aa**, and the feminine form in –**oo**.

Example:

guddaa (m.) **nama guddaa** a big man

guddoo(f.) **nama guddoo** a big woman

b. In the second group the masculine form terminates in –**aa**, the feminine form in –**tuu** (with different assimilations).

Example: **dheeraa**(m.) **nama dheeraa** a tall man

dheertuu(f.) **intala dheertuu** a tall girl

c. Adjectives that terminate in –**eessa** or **–(a)acha** have a feminine form in –**eettii** or –**aattii**.

Example: **dureessa** (m.) **nama dureessa** a rich man

dureettii (f.) **nitii dureettii** a rich woman

d. Adjectives whose masculine form terminates in a long vowel other than –**aa** as in short vowel –**a** (but not of the suffix **–eessa/-aacha)** are not differentiated with respect to their gender.

collee(m.) **farda collee** an active horse

collee(f.) **gaangee collee** an active mule

II. Number

There are four groups of adjectives with respect to number. These are:

a. Most of the adjectives form the plural by reduplication of the first syllable masculine and feminine adjectives differ in plural as they do in singular [47].

Example:

Singular	Plural
guddaa(m.)	**guguddaa**(m.)
guddoo(f.)	**guguddoo**(f.)
xinnaa(m.)	**xixinnaa**(m.)
xinnoo	**xixinnoo**

pl.f. **lageewwan guguddoo** big rivers

pl.m. **qubeewwan guguddaa fi xixiqqaa** big and small letters

b. There is a further plural form which is gender neutral for adjectives of this group beside a special masculine and feminine plural. This plural form terminates in **-oo**, and is sometimes used with reduplication and sometimes without. Table 4 shows examples of plural adjectives formed by reduplication which are gender neutral

Singular	**plural**	**plural**
Jabaa (M)	**Jajabaa**(M)	**Jajjaboo**(Gender neutral)
Jabduu (F)	**Jajjabduu**(F)	

c. Adjectives which may function as nouns as well form the plural only by using noun plural suffixes. Table 5 shows examples of plural adjectives formed using noun plural suffixes

Singular		**Plural**	
M	F	M	F
Dureessa	Dureettii	Dureeyyii/dureessota	dureettiwwan

d. Adjectives of the fourth group form the plural without marking the gender, very often by reduplication of the first syllable. Sometimes adjectives of this group form the plural by using a noun plural suffix [19].

Singular	**Plural**	**English**
Adii	a`adii/adaadii	White
Collee	Colleewwan	Active

III. Definiteness

The demonstrative pronouns that express definiteness in Afaan Oromo follow the adjective if the noun is qualified by an adjective and a demonstrative pronoun as well.

Example: **Namicha dheeraa sana argitee**? Did you see that tall man?

The suffix –**icha** that sometimes has a definite function normally is suffixed to nouns, but it can be suffixed to adjectives or numerals, too,

Example **Lagni guddichi** the big river **namichi tokkichi** a single man

IV. Compound Adjectives

In the new terminology of Afaan Oromo compound adjectives play a growing role.

Example: **afrogaawaa** **afur + rogaawaa** rectangular four + angled

sibilala **sibila + ala** non-metal metal + outside

3.2.4 Adverbs

Adverbs have the function to express different adverbial relations such as relations of time, place, and manner or measure.

Some examples of adverbs of time:

amma now

booda later

Some examples of adverbs of place:

achi(tti) there

ala outside

Some examples of adverbs of manner:

saffisaan quickly

sirritti correctly

Some examples of adverbs of measure:

baay'ee , **danuu** much , many , very

duwwaa only, empty

3.2.5 Pre-, Post, and Para-positions

Afaan Oromo uses prepositions, postpositions and para-positions [46].

I. Postpositions

Postpositions can be grouped into suffixed and independent words.

Suffixed postpositions

-tti in, at, to

-rra/irra on

-rraa/irraa out of, from

The post position –**tti** is used to form the locative. The postposition **-rraa/irra** may be used to express a meaning similar to ablative.

Example: **Adaamaatti yoom deebina?** When shall we go back to Adama?

Gammachuun sireerra ciise. Gemachu lay down on bed.

Post position as independent words

ala outside **wajjiin** with, together with

bira beside **teellaa** behind

Example: Namoota nu bira jiraniis hin jeeqnu. We don't hurt people who are with us.

II. Prepositions

akka like, according to

gara to, in the direction of

hanga/hamma until, up to

karaa along, the way of, through

The prepositions **gara**, **hanga**, and **waa'ee/waayee** are still treated as nouns and therefore are used in a genitive construction with other noun they belong to, expression: the direction to, the matter of, etc.

Example:

Namni akka harkaan waa hojjechuuf fayyadamu argi maalitti fayyadamaa? As people use hands to work something what does the elephant use?

III. Para-positions

Gara… tti to **Gara… tiin** from the direction of

Example: **Lukkichi rifatee jeedaloo dheesuuf gara manaatti gale.** The cock was scared and went home to take refuge from the fox.

3.2.6 Conjunctions

Conjunctions are unchanging words which coordinate sentences or single ports of a sentence. The main task of conjunctions is to be a syntactical formative element that establishes grammatical and logical relation between the coordinated constituents.

According to [46] the main functions of conjunctions are identified as: the function of coordinating clauses (coordination), the function of coordinating parts of sentence (coordination) and the function of coordinating syntactical unequal clauses(subordination). On the other hand, with regard to their form we can subdivide the conjunctions of Afaan Oromo into:

I. Independent Conjunctions

a. Coordinating

Example: **garuu** but

Hoolaan garuu rooba hin sodaattu. But the sheep is not afraid of rain.

b. subordinating

Example: **akka** that, as if, as whether

Maaliif akka yaada dhuunfaa yookaan yaada haqaa akka ta'e adda baasii barreessi.

Write separately why it is an individual opinion or that it is an opinion about justice

II. Suffixed Conjunctions

Example: **–f/ -fi/ -dhaaf** and, that, in order to, because, for

uffata uffachuuf bittee? Did you buy the clothe for wearing?

III. Conjunction consisting of one, two or more parts

Conjunctions consisting of two parts can be formed by two independent words or two enclitics or one independent word plus enclitic. They can be formed made up of two single conjunctions that are used after each other in order to give more detailed information about the logical relation or to intensify it.

Example: **akkam akka** how, that

Dura namni tokko beekumsa mammaaksaa akkam akka jabeeffatu ilaaluu nu barbaachisa. At first, we have to see how a person extends the knowledge of proverbs

IV. Conjunctions consisting of several segments

Conjunctions consisting of several segments are copulative or disjunctive conjunctions which as they stand separately from each other are to emphasize the segments of a parallel construction. These are stable, stereotyped constructions the first segment of which has to be followed by a certain second segment:

Example: **–s… -s,** as well as

Jechoota hudhaa wajjiniis, hudhaa malees karaa lamaan barreeffaman

Words with glottal stop as well as without glottal stop are written in two ways.

3.3 Word and Sentence Boundaries

In Afaan Oromo, like in English the blank space shows the end of a word. Moreover, parenthesis, brackets, quotes, etc. are being used to show a word boundary. Sentence boundaries punctuations are also similar to English language i.e., a sentence may end with a period (.), a question mark (?), or an exclamation point (!) [47].

Morphology adds a burden to NLP works. For the purpose of text summarization and also other NLPs, the variant words of a morpheme should be reduced to their root so that they can be counted as one while calculating term frequency, and in our case when creating a Word2Vec model. Using stemmer is believed to minimize the difficulty of dealing with different forms of a word [47]. There have been efforts of developing stemming algorithm for Afaan Oromo. We used the algorithm developed by [47] for our work.

Chapter Four: Related Work

In this Chapter, we describe researches focusing on a single document in news domain applying different techniques. We first focus on reviewing some of works for non-Ethiopia, and then review local works in the area of text summarization. Summary also given at the end.

4.1 Text Summarization for non-Ethiopian Languages

Text summarization was introduced 50 years ago by Luhn [9] which was conducted for English language. The author proposed that words appearing many times in a text carry a good idea about the content of the document though there are words that appear very frequently but not content bearing. As a result, the researcher tried to cut off these words by determining a fixed threshold. The idea of Luhn was acknowledged and used in many automatic information processing systems. The system developed takes a single document as input. It is domain-specific to summarizing technical articles and the system used features like term filtering and word frequency (low-frequency terms are removed). Sentences are weighted by the significant terms they contained and sentence segmentation and extraction are performed.

Edmundson [10] expanded the work of Luhn [9] for the same language. The author carefully outlined the human extracting principles and noticed that the location of a sentence in a text gives some clues about the importance of the sentence. Thus, the researcher suggested word frequency, cue phrases, title and heading words, and sentence location as an extraction feature. Like the work of Luhn, Edmundson's system is a single document and domain-specific (that deals with technical articles). Moreover, the output of the system is an extracted summary. Since then many systems have been developed in the area of automatic text summarization both on single and multi-documents.

Padmakumar and Saran [11] came up with an unsupervised technique to summarize text. The authors proposed for both extractive and abstractive methods of text summarizations. They have used sentence embeddings to detect paraphrases for text summarization. To obtain a sentence embedding, the authors combined word embeddings that satisfy the property that sentences that are paraphrases of each other embedded near each other in the vector space. They proposed to cluster sentences projected to a high dimensional vector space to identify groups of sentences that are semantically similar to each other and select representatives from these clusters to form a summary. The extractive method simply chooses sentences from the

text whose embedding is the nearest, in terms of Euclidean distance, to the centroid of the cluster. In the abstractive method, a decoder is trained to decode embeddings into sentences. They used a recurrent neural network with long short-term memory to encode embeddings into sentences. The authors used paragram and skip-thought to create the embeddings and they used k-means and mean shift for clustering. According to their experiment, the result of extractive summarization is better than abstractive. They achieved 0.4141 precision for extractive which used paragram for embedding and k-means for clustering. They have got 0.3366 precision for extractive which used skip-thought for embedding and k- means for clustering. For abstractive summarizations, they achieved less than 0.29 by exchanging the above methods.

Samer *et al*. [12] proposed unsupervised multi-document Arabic Text summarization which is based on clustering and Word2Vec. The authors used Word2Vec to map the words to fixed-length vectors and, to obtain the semantic relationship between each vector based on the dimensions. They used K-means algorithm for two purposes: selecting the distinctive documents and tokenizing these documents to sentences, and using another iteration of the k-means algorithm to select the key sentences based on the similarity metric to overcome the redundancy problem and generate the initial summary. Lastly, the authors used weighted principal component analysis (W-PCA) to map the sentences' encoded weights based on a list of features. This selects the highest set of weights, which relates to important sentences for solving incoherency and readability problems. Their work has six main stages: data collection, text preprocessing, selecting the discriminative documents for generating the initial summary, sentence tokenization, sentence weight mapping, and selecting sentences based on the best weight as the final summary. The final step is evaluation. Their system registered F-score of 0.644.

Gaetano *et al*. [13] proposed a centroid-based method for text summarization that exploits the compositional capabilities of word embeddings. The authors adapted the centroid-based method by introducing a distributed representation of words where each word in a document is represented by a vector of real numbers of an established size. Formally, given a corpus of documents [D1, D2, . . .] and its vocabulary V with size N = |V|, they defined a matrix, so-called lookup table, where the i-th row is a word embedding of size k, k< N, of the i-th word in V. The values of the word embeddings matrix are learned using the neural network model. In

order to build a centroid vector using word embeddings, they first select the meaningful words into the document. For simplicity and a fair comparison with the original centroid method, they selected those words having the TF-IDF weight greater than a topic threshold. Thus, they compute the centroid embedding as the sum of the embeddings of the top ranked words in the document using the lookup table. To get the sentence score, the authors used cosine similarity between the embedding of the sentence and that of the centroid of the document. Finally, the top ranked sentences are iteratively selected and added to the summary until the limit is reached. The authors achieved 38.81% Rouge 1 and 9.97% Rouge 2 value.

In [14], Seq2Seq models have been used for eBay product description summarization. The authors came up with Document-Context based Seq2Seq models using RNNs for abstractive and extractive summarizations. The authors used the idea that humans understand a document by only reading the tittle, abstract or any other contextual information before reading the document to propose that Seq2Seq models should be started with contextual information at the first-time step of the input to obtain better summaries. In this manner, the output summaries are more document centric, than being generic, overcoming one of the major hurdles of using generative models. They generated document context from user-behavior and seller provided information. They trained and evaluated their models on human-extracted-golden-summaries. The document-contextual Seq2Seq models outperform standard Seq2Seq models

4.2 Text Summarization for Ethiopian Languages

4.2.1 Text summarization for Amharic Language

Kamil Nuru [15] developed a single document Amharic summarization. The author used surface level statistical features to assign weights to sentences. The highest-scoring sentences were extracted to form the summary. Seven features are used: title words, cue phrases, the first sentence of the document (header), words in a header, first sentence of a paragraph, paragraph end sentences, and high-frequency words (keywords). Each feature has an associated weight, obtained from training with manual summaries of four news articles, and the weights are combined linearly to produce an overall score for a sentence. As part of a learning phase, the stop words and cue phrases list of the system can be updated by a user that is generating a summary. They achieved 58% Recall and 70.4% Precision.

Melese Tamiru [18] proposed two methods that can rank and extract sentences to be included in the summary of the Amharic document. The first method employs Latent Semantic Analysis (LSA) along with the document genre information to select semantically important sentences. The second method combines latent semantic analysis with graph-based ranking algorithms to compute the relevance of sentences to be included in the summary. The authors achieved 0.42 F-measure at 20% compression rate and 0.47 F-measure at 30% compression rate.

Eyob Delele [16] investigated the problem of building a concept-based single-document Amharic text summarization system. Because Ethiopan languages like Amharic lack extensive linguistic resources, the author proposed to use a statistical approach called topic modeling to create text summarizer. More specifically, the author proposed to use the topic modeling approach of probabilistic latent semantic analysis (PLSA). The authors show that a principled use of the term by concept matrix that results from a PLSA model can help produce summaries that capture the main topics of a document. The researcher proposed and tested six algorithms to help explore the use of the term by concept matrix. All of the algorithms have two common steps. In the first step, keywords of the document are selected using the term by concept matrix. In the second step, sentences that best contain the keywords are selected for inclusion in the summary. To take advantage of the kind of texts, the author experimented with news articles. The algorithms always select the first sentence of the document for inclusion in the summary. The authors evaluated the proposed algorithms for precision/recall for summaries of 20%, 25% and 30% extraction rates. The best results achieved are as follows: 0.45511 at 20%, 0.48499 at 25% and 0.52012 at 30%.

Kifle Deresse [17] introduced graph-based Automatic Amharic Text Summarizer (GAATS), a generic and domain-independent graph-based model for automatic single-document summarization task, and shows how this model can successfully be used to generate extracts of high quality from Amharic texts. In particular, the author extended the two prominent graph-based link analysis algorithms: PageRank and HITS with two-sentence centrality measures: cumulative sum and discounted cumulative sum for exploiting the relation between sentences in a text and/or node in a graph, and showed the results of their experiments. The results demonstrated that extractive summaries of better quality can be generated when discounted cumulative sum paired with HITS. The results also revealed that the researchers'

approach is domain-independent and more effective than reference summarization systems. The authors achieved 0.632 F-measure at 20% compression rate and 0.697 F-measure at 30% compression rate.

4.2.2 Text summarization for Afaan Oromo

Girma Debele's [19] work is the most notable work conducted in 2012. The author proposed three methods. The first method (M1) uses term frequency and position methods without Afaan Oromo stemmer and other lexicons (synonyms and abbreviations). The second method (M2) is a summarizer with a combination of term frequency and position methods with Afaan Oromo stemmer and language-specific lexicons (synonyms and abbreviations) and the third method (M3) is with improved position method and term frequency as well as the stemmer and language-specific lexicons (synonyms and abbreviations). The performance of the summarizers was measured based on subjective as well as objective evaluation methods. The result of objective evaluation shows that the three summarizers: M1, M2, and M3 registered f-measure values of 34%, 47%, and 81% respectively i.e., M3 outperformed the two summarizers (M1 and M2) by 47% and 34% respectively. Moreover, the subjective evaluation result shows that the three summarizers' (M1, M2, and M3) performances with informativeness, linguistic quality, and coherence and structure are: (34.37 %, 37%, and 62.5%), (59.37%, 60%, and 65%), and (21.87%, 28.12%, and 75%) respectively as it is judged by human evaluators. In both subjective and objective evaluation, the results are consistent. Summarizer M3 that uses the combination of term frequency and improved position methods outperformed other summarizers followed by M2.

Fiseha Berhanu [8] developed Afaan Oromo text summarization based on the research gap found in [19] and also used open text summarizer (OTS) open source. The author's work is a generic Afaan Oromo news text summarization based upon sentence selection functions. The researcher used features like sentence position, keyword frequency, cue phrase, sentence length handler, the occurrence of numbers and events like time, date, and month in sentences to develop the system. The author has got different results by combining these features and making a different combination of these features. The system has been evaluated based on seven experimental scenarios and evaluation is made both subjectively and objectively. The subjective evaluation focuses on the evaluation of the structure of the summary like referential

integrity and non-redundancy, coherence, and informativeness of the summary. The objective evaluation uses metrics like precision, recall, and F-measure evaluation. The result of the subjective evaluation is 88% informativeness, 75% referential integrity and non-redundancy, and 68% coherence. Because of the added features, different techniques, and experiments applied to this work the system gave 87.47% F-measure and outperform by 26.95% than in [19].

Asefa Bayisa [20] focused on developing query-based Afaan Oromo text summarization. The author's work makes its central point a user query to come up with the summary. The author also focused on the research gap found in [19] and tried to change the technique from generic to query-based to increase the performance of the summarizer. The results of the evaluations showed that the proposed system registered f-measure of 82%, 78%, and 82% at a summary extraction rate of 10%, 20%, and 30% respectively when VSM is used along with position method. Moreover, the informativeness and coherence of the proposed system also registered its best performance summary of 59%, 77%, and 91% average score on five scale measures at an extraction rate of 10%, 20%, and 30% respectively when both methods are used together.

4.3 Summary

From the above discussion, we can see that many works have been done by using various techniques. There are no common techniques that are used in all languages. Depending on the gaps found, in this work we focus on summarization of different Afaan Oromo texts using word embeddings as feature enrichment. We hypothesize that word embedding features may or may not boost the performance of Afaan Oromo text summarizer. As a gap we did found two major weaknesses. The first one is, because of the difference in morphological structure we cannot directly apply the works conducted on non-Ethiopian languages for Afaan Oromo. Secondly, the researches conducted on Afaan Oromo are based on sentence selection functions, and they lack coherence and cohesion. Also, they are found to be redundant. In order to overcome these, we employ word embedding features which can handle the semantic relations of words depending on their vector representations. In addition, we are going to use suitable algorithms and techniques that may or may not increase the performance of the Afaan Oromo text summarizer. Besides, in this work we are going to see the performances difference between this work and the previous work.

Chapter Five: Design and Implementation of Afaan Oromo Text Summarizer

In this Chapter, we present how Afaan Oromo text summarizer is designed and implemented based on word embedding and graph-based algorithms.

Almost all extractive text summarization shares three core phases: preprocessing, extraction and summary generation. In each phase there are list of works which has to be done to get the desired summary. The architecture of our work has 4 main components, namely, preprocessing, generating sentence vectors from pre-trained word embeddings, creating a graph representation, and finally, sentence weight mapping, and selecting sentences based on the best weight as the final summary. The architecture is depicted in Figure 5.1.

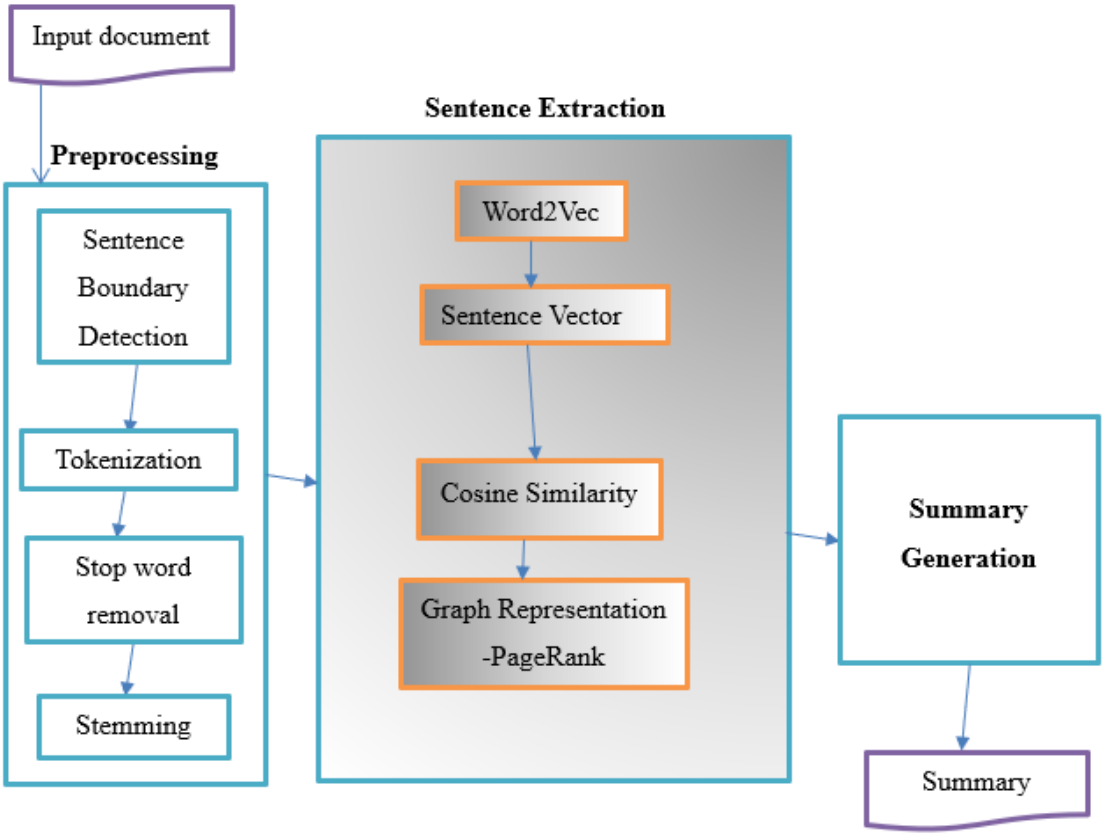

Figure 5. 1: Architecture of the summarizer

5.1 Preprocessing

In preprocessing phase, there are four basic components which have been used in our work. Each component has their own task and algorithms. The components and their algorithms are described as follows.

5.1.1 Sentence Boundary Detection

Sentence boundary detection is the problem of deciding where sentences end. There is a straightforward solution in many cases (i.e. in most cases, splitting after a period or question/exclamation mark is the right decision). However, this is not true for the last sentence: abbreviations, quotations including punctuation, and many other situations make this task far from trivial. In Afaan Oromo sentences end with period, exclamation/question mark like English. But as described above there are many conditions which make this false. There are many abbreviations in Afaan Oromo which have period in between but, they do not indicate the end of the sentence. For Example, "**k.k.f**" is an abbreviation of "**kan kana fakkaatan"**. So, in order to solve this, we have gathered Afaan Oromo abbreviations from different sources [8, 20, 47] and added a regular expression to the rule. See Appendix A for list of abbreviations gathered. The detector works according to Algorithm 5.1

```
Input: Afaan Oromo text, and abbreviation lists
Output: List of sentences after splitting them
Let sent be a string which set to be null
Let ablist is a list of abbreviations
       Read file
       Convert each character in file to lower case
For abrev in ablist
       Read list
       If file contains list
              Remove the word from the list
        If file in read endswith '.', '!', or '?'
                 Put each sentence in sent
Return sent
```

Algorithm 5. 1: Sentence Boundary Detection Algorithm

5.1.2 Tokenization

Word tokenization is the problem of splitting a sentence into separate word tokens. While splitting on whitespace is an excellent heuristic, this approach fails in many cases. In Afaan Oromo there are many conditions that have to be considered when tokenizing a word. One of such conditions is the use of single quote /‘/. For example, **bu’e** is a single word. To tokenize this word as a single word we have added regular expressions into the NLTK tokenizer. The tokenizer works according to Algorithm 5.2.

```
Input: Afaan Oromo text, and abbreviation lists
Output: Tokenized sentences
Let sent be a string which set to be null
Let tokens be a list which set to be null
       Read file
       Convert each character in file to lower case
For file in sent
       Split each word using white space
       Put each word in tokens
If tokens contains “?,$#@^&*1234567890
     Replace tokens with white space
Return tokens
```

Algorithm 5. 2: Tokenization Algorithm

5.1.3 Stop Word Removal

In our research we used NLTK stop word remover. We gathered more than 200 stop words from different sources including the once gathered by [8, 19]. The entire list of stop words gathered are available in Appendix B. The remover works according to Algorithm 5.3.

```
Input: tokenized text and a list which contains stop words
Output: stop word free text
for each sentence in each paragraph
      if a word in each sentence is stop word
             remove the word from the sentence
  end for
return list of sentences without stop word
```

Algorithm 5. 3: Stop word removal algorithm

5.1.4 Stemming

Stemming is the process of reducing the inflected forms of a word to its root form by stripping off the affixes. We used the algorithm developed by [47] for our work.

5.2 Sentence Extraction

In our research to generate the final score, the system calculates the scores iteratively according to the rules of graph-based algorithms. The basic idea of iterative calculation process based on word-sentence relationship and graph model is that a sentence which has a more keywords has higher weight, and a word which occurs more frequently in a high weighted sentence can get high weight.

To get the best sentences from the given text we have used four main features: Word2Vec, sentence vector, cosine similarity, and graph representation.

5.2.1 Word2Vec

Training a Word2Vec needs a large amount of data. The bigger the data is, the better the representation will be. In Afaan Oromo there is no pre-trained Word2Vec, and also there is no collected set of documents that will be used to train this embedding. So, to get the pre-trained word embeddings we have gathered different documents from different sources. Since, our work is a text summarization which will be news obviously, building a good corpus is mandatory. We collected over 1000 news topics from different news portals like BBC/AfaanOromoo, VOA/AfaanOromoo, Kallacha Oromia, etc… And to get a bigger collection of words we have tried to collect different Afaan Oromo documents from different individuals and organizations. By collecting these sets of large documents, we have used a python library called “Gensim” to train the Word2Vec. The word to vector model which is used to create the word vector works according to Algorithm 5.4.

```
Input: d: dataset
Output: Matrix W(256,300) of one-hot vectors for each possible byte
value(0-255)
Let f be a list of tuples (byte_value, frequency)
for i: =0 to 255 do
   freq =0
   for each item j in d do
```

```
     freq=freq + frequencyOfOccurence(i, j)
   end for
    append (i, freq) tuple to f
end for
f=sort f based on frequency
w=word2vec(f,300)
return W
```

Algorithm 5. 4: Algorithm for creating word vector

5.2.2 Sentence Vector

To get a sentence vector, which will be used as input for measuring the similarity of the sentences we used a pre-trained Afaan Oromo Word2Vec. After we get this word representation vector, we used it to get a vector representation of the document to be summarized. To calculate the sentence vector, we used the most commonly used average Word2Vec. We represented every word in a sentence as a 300-dimensional vector using the pre-trained Word2Vec vectors. We then calculated the sentence score of this feature by taking the mean of each dimension of all the word vectors, forming a vector. The sentenced vector works according to Algorithm 5.5.

```
Input: Afaan Oromo sentences and Word2Vec model
Output: Sentence vector
Let sent be a sentence
Let vec be a vector
if the length of sent==0
   return 0.0
vec=model of a word in sent
for word in sent
        vec = vec + model of the word
end for
return vec/length of sent
```

Algorithm 5. 5: Algorithm for calculating sentence vector

5.2.3 Cosine Similarity

To calculate the similarity between sentences we used the cosine value of the angle between them. The cosine similarity between the words is calculated as Equation 3 [44].

$$\boldsymbol{sim(X,Y)} = \frac{\vec{X} * \vec{Y}}{\left\|\vec{X}\right\| * \left\|\vec{Y}\right\|} \quad \textbf{(3)}$$

Where X, Y are the two sentences and $\underset{X}{\rightarrow} and \underset{Y}{\rightarrow}$ are the sentence vectors. The similarity is measured according to Algorithm 5.6.

```
Input: sentence vectors
Output: cosine similarity
Read vectors
Calculate similarity
Return similarity
```

Algorithm 5. 6: Algorithm to calculate cosine similarity

5.2.4 Graph Representation

In our research an undirected graph G is established in which the sentence is treated as a node and the similarity between sentences is treated as edge using sentence similarity matrix, where each row and columns in the matrix correspond to a particular sentence that represent node in the graph and each cell values represent similarity between the corresponding sentence pair that establish an edge between nodes in the graph. We calculated the sentence similarity matrix using the cosine similarity value obtained in Equation 3. To compute the score of each node on the graph G or sentence in text we used centrality measures. The centrality measures computes sentence's centrality using the mean of link weights of the sentence with others considering links whose weights is greater than or equal to specified threshold. Then, set the corresponding row and column values of the matrix related to that sentence to zero, and compute the centrality of the next sentence based on the contributions made by the remaining 'n-1' sentences… etc., this process iterates until the centrality scores of all sentences are obtained. Thus, centrality score of each node on the graph G is computed as Equation 4 [17]:

$$ai = \left(\frac{1}{N-1}\sum_{i=1}^{k} wij\right) \quad (4)$$

Where ai denotes the centrality of sentence i, wij is the cosine similarity of sentences si and sj, and N is the number of sentences in the text or nodes in the graph.

After building the graph G = (V, E), we compute the salient score for each node on graph G using PageRank, and then rank them according to their degree of importance. The PageRank equation used to rank the sentences is defined as Equation 5 [33].

$$\boldsymbol{Si} = \frac{\mathbf{1-d}}{\boldsymbol{m}} + \boldsymbol{d} * \sum_{\boldsymbol{d\epsilon In(di)}} \boldsymbol{Sj} * \frac{\boldsymbol{sim(di,dj)}}{\sum_{\boldsymbol{dk\epsilon Out(dj)}} \boldsymbol{sim(dj,dk)}} \quad (5)$$

Where Si is the weight of i^{th} sentence, d is the damping coefficient (default is 0.85), m can be 1 or the number of the sentences in the article. The impact of other sentences on the current sentence is controlled by d and m. d_i is a sentence node in graph G, which represents the i^{th} sentence in S. $In(d_i)$ represents the set of nodes pointing to the sentence node d_i, and $Out(d_j)$ represents the set of nodes pointed to by d_j. S_j is the weight of the sentence node d_j in this round.

5.3 Summary Generation

This module will be activated after the sentences are extracted and ranked according to their score. The final task of our system is generating the top N selected sentences which can represent the whole document. N is a compression ration defined based on how much words or sentences in general we need to see in the final summary. The selected sentences are then rearranged to increase the readability and coherence. If the sentence selected contains the removed stop words tokens and abbreviations, it will be appended to the sentence. Generally, it selects the best combination of sentences that gives the important information of the original text and the desired summary is the combination of top N important sentence. The optimal and best collection of sentences is selected to maximize overall importance, minimize redundancy, or maximize coherence in global selection procedure. To get the summary, sentences which have a higher similarity to the sentence with the highest weight than a threshold is regarded as repetitions and omitted. Ultimately, summaries can be generated in the top N sentences, and the N can be determined according to some conditions like the number of summaries and the number of words in summary /the number of text words. In our case we used compression ratio to generate the number of sentences included in the final summary. We used 30% compression ratio. The summarizer works according to Algorithm 5.7.

```
Input: Afaan Oromo sentences ranked by descending order
Output: Top N selected sentences (final summary)
Let say compression ratio is set to 30%
Let sent is the sentences in the original document
Let sent_score is a list of sentences ranked by descending order
Begin
  N= length of sent multiplied by 0.3
  selected_sent=0
  While selected_sent <=N
      finalSummary=N top sentences from Sent_score
      selected_sent=selected_sent +1
  Return finalSummary
End
```

Algorithm 5. 7: Summary Generation Algorithm

Chapter Six: Experimental Result and Analysis

In this Chapter, the performance of the summarization system is discussed. In addition to that we discuss how corpus is prepared for system development and for testing the system.

6.1 Corpus Preparation

As stated in Section 1.5 we gathered data of two categories: the data used to develop the pre-trained model and the data used for experimentation (validation and testing data including lexicon data). To develop the pre-trained model, data of different domains have been gathered from different sources. We started collecting the data needed to develop the model from scratch because, as far as our knowledge there is no researches conducted using word embeddings for Afaan Oromo. The Word2Vec model needs millions of words for training to give a good result. Getting these much data is not simple for Afaan Oromo. The 70% of our training data is collected from different organizations and individuals. We also accessed different online news portals like BBC/AfaanOromoo, VOA/AfaanOromoo etc. to get the rest of our training data. The collected data will be around one million words approximately.

The data for experimentation is also collected from online news portals. To collect balanced corpus for our system, we considered different topics like health, sport, politics, technology, art, education etc. Table 6.2 shows the statistics of the experimentation corpus. The corpus average number of words is **416** approximately, number of sentences is **24** approximately and **8** paragraphs approximately.

We have selected **22** different documents as discussed earlier. Number of selected articles from each topic is listed in Table 6.1.

Table 6. 1: Number of selected articles from each topic

Topics	Number of selected topics
Art	3
Education	3
Health	3
Metrology	3
Sport	4
Technology	3
Politics	3

Table 6. 2: Statistics of experimentation data

Topic ID	News size in words	News size in sentence	News size in paragraph
Topic 1	529	25	7
Topic 2	599	30	11
Topic 3	566	31	11
Topic 4	564	36	11
Topic 5	271	18	6
Topic 6	601	36	13
Topic 7	234	18	4
Topic 8	204	13	3
Topic 9	628	50	14
Topic 10	279	13	7
Topic 11	397	22	11
Topic 12	539	32	10
Topic 13	801	34	12
Topic 14	283	16	7
Topic 15	318	13	5
Topic 16	224	15	6
Topic 17	206	12	5
Topic 18	200	12	5
Topic 19	400	20	8
Topic 20	206	14	4
Topic 21	357	22	9
Topic 22	743	39	11
Average	**415.86**	**23.68**	**8.18**

6.1.1 Reference Summary Preparation

The reference summary is a summary prepared by human experts to test the performance of the system. We have selected **13** topics randomly from topics listed in Table 6.2. These topics are distributed to **5** Afaan Oromo Instructors from Wollega university. The respondents were expected to rank the sentences according to their importance for inclusion in the summary. We have used 40% compression ratio as shown in Appendix E. Therefore, the respondents gave the rank for the sentences until the number of sentences needed are selected. For example: if the given topic has 12 sentences the summary should contain 5 sentences (12*0.4=4.8 ≈**5** by rounding off to the nearest integer). The result of reference summary was used for objective evaluation.

6.1.2 System Summary Testing Data Preparation

This data is used for subjective evaluation. For this purpose, we have used the rest **9** topics only because, in this case the respondents are expected to evaluate the summary generated by the system which needs more attention. We have prepared a guide which the respondents used while evaluating the system summary. See Appendix F for the guide lines.

6.2 Evaluation and Discussion

We conducted both subjective and objective evaluation. The discussion also made for both subjective and objective evaluation.

6.2.1 Subjective Evaluation

Subjective evaluation method requires human judgements based on categories like informativeness, non-redundancy, grammar, referential clarity and coherence. In this research we have generated summaries. The generated summaries are evaluated by subject evaluators. The subject evaluators evaluate the performance of the system based the following three points.

- ✓ The summary informativeness
- ✓ Grammar, Non-redundancy and referential clarity
- ✓ Structure and Coherence

We requested the subject evaluators to read the original documents carefully before evaluating the summary. In order to resolve miss understanding during evaluation we have prepared a description of these three points on the questionnaire as shown in Appendix F.

A. Summary Informativeness

We have generated the summary of the documents and the result of the summary is collected from the subject evaluators and discussed as follows. The scoring mechanism used in this study is based on 5 different point; Very Good= 5, Good=4, Not bad= 3, Poor= 2 and Very Poor =1. Each topic is evaluated by four different subject evaluators. The result of the informativeness of the summary is shown in Table 6.3. The score is out of 100%, the percentage of the informativeness of the summary for each experiment is the sum of the score given by each subject evaluator for each topic, divided by the sum of maximum score.

$$\frac{\sum_{i}^{4} Ri}{20} * 100 \qquad (3)$$

Where, *Ri* is result scored by each subject evaluator.

For example, using this equation the result of ***Test 2*** shown on Table 6.3 is (5 +4 +4 + 5) = (18/20) *100= *90%*. The result obtained is shown in Table 6.3.

Table 6. 3: Result of informativeness of the summary

Test/DocId	**Result**
Test 2	90%
Test 4	80%
Test 9	90%
Test 12	90%
Test 13	60%
Test 18	90%
Test 19	100%
Test 21	70%
Test 22	80%
Average	**83.33%**

The information preserved with stemmer and language specific lexicons **83*.33%.*** The stemmer improved the result because, when we use stemmer each word is inflected to its root form. This makes number of words to be represented by single root and this helps to get the semantic relatedness of words simple for our Word2Vec model. Semantic relatedness in return, helps to avoid redundancy and select semantically non-related sentences from the original document to improve the informativeness of the summary. Depending on this result we can deduce that the informativeness of a summary improved when stemmer and Language specific lexicons is combined together with word embedding.

B. Non-redundancy and Referential Clarity

In this point the subject evaluators give a score depending on three points. First, they check whether the summary contains unnecessary repetition or not. Secondly, the respondents check for proper usage of grammars. Lastly, they check for referential clarity, to evaluate a proper usage pronouns and noun phrases. We have followed the same scoring mechanism used for informativeness. The result obtained is shown in Table 6.4.

Table 6. 4: Non-redundancy and referential clarity

Test/DocId	Result
Test 2	90%
Test 4	70%
Test 9	80%
Test 12	90%
Test 13	70%
Test 18	100%
Test 19	90%
Test 21	60%
Test 22	60%
Average	**78.88%**

According to the result obtained the summarizer achieved ***78.88%*** grammar, non-redundancy and referential integrity.

C. Structure and Coherence

Another subjective evaluation is a coherence of the summary. In this point we let the subject evaluators to measure the smooth transition of sentence from sentence for generated summary. We used the same scoring mechanism for this too and the result obtained is shown in Table 6.5.

Table 6. 5: Result of Structure and coherence evaluation

Test/DocId	Result
Test 2	80%
Test 4	70%
Test 9	70%
Test 12	80%
Test 13	60%
Test 18	90%
Test 19	90%
Test 21	80%
Test 22	70%
Average	**76.66%**

The structure and coherence of the summary is ***76.66%*** when we combined language specific lexicons and stemmer with word embedding in the summarizer. In general, our system scores ***83.33%*** informativeness, ***78.8%*** referential integrity and non-redundancy, and **76.66%** structure and coherence.

6.2.2 Objective Evaluation and Discussion

Objective evaluation needs a reference/golden summary to evaluate the system generated summary. To get the reference summary we contacted 5 Experts as stated in Section 6.1.1. For this evaluation we used 13 different topics which gathered from different sources. The result obtained is shown in Table 6.6.

Table 6. 6: Objective evaluation result of the summarizer

Test/DocId	Result		
	P	R	F-Measure
Test 1	0.444	0.4	0.42
Test 3	0.6215	0.5765	0.598
Test 5	0.2495	0.2135	0.231
Test 6	0.4375	0.35	0.3885
Test 7	0.7495	0.5625	0.6426
Test 8	0.775	0.58	0.663
Test 10	0.5	0.6	0.545
Test 11	0.333	0.333	0.333
Test 14	0.5	0.5	0.5
Test 15	0.4995	0.375	0.428
Test 16	0.5	0.4165	0.454
Test 17	0.75	0.283	0.409
Test 20	0.5	0.3	0.375
Average	**0.527**	**0.422**	**0.468**

The result obtained from objective evaluation is 0.527 precision. 0.422 recall and 0.468 F-measure.

6.2.3 Afaan Oromo Text Summarization using Word Embedding vs Afaan Oromo Text Summarizer (AOTS)

AOTS is the work of Fiseha Berhanu [8] and is conducted on Afaan Oromo and used some common features like the ones we used in this study. The authors work expanded the work of Girma [19] by adding features and the authors achieved promising results. As the author stated his work outperformed the previous one. So, to evaluate the performance of this study we compared with the work of Fiseha [8]. To measure the gap, we have used similar data which is used in their work and the same compression ratio (CR). As illustrated in Table 6.7 the performance difference

among Afaan Oromo Text Summarization using word embedding and AOTS is **0.052** Precision, **0.064** Recall and **0.058** F-measure. AOTS achieved greater result.

Table 6. 7: Result of Word embedding vs AOTS

Test/DocId	**Word Embedding**			**AOTS**		
	P	R	F-Measure	P	R	F-Measure
Test 1	0.5	0.5	0.5	0.764	0.725	0.744
Test 2	0.745	0.745	0.745	0.766	0.75	0.758
Test 3	0.7	0.6335	0.665	0.766	0.6	0.673
Average	**0.648**	**0.626**	**0.637**	**0.7**	**0.69**	**0.695**

6.3 Summary

In this thesis work we used word embedding with graph-based algorithms. Our work achieved *0.648* Precision, *0.626* Recall and *0.637* F-measure with the data used in the work of [8]. When we compare our work with [8], our work outperformed by *0.052* Precision, *0.064* Recall and *0.058* F-Measure. Based on the results, combining word embedding with graph-based algorithms does not help to improve the performance of Afaan Oromo Text Summarization. To improve the results, we suggest changing graph-based algorithm to neural network algorithms.

Chapter Seven: Conclusion

7.1 Introduction

The goal of this study was developing and evaluating automatic text summarizer for Afaan Oromo text. We have used word embedding for feature enrichment and PageRank to rank the sentences. Language specific lexicons and stemmer are used alongside word embedding in this study. We conducted both subjective and objective evaluations and promising result has been obtained by combining word embedding with stemmer and language specific lexicons. In this chapter conclusions, recommendations and future work are presented based on the findings of the study.

7.2 Conclusion

In this master's thesis word embedding is used as a feature enrichment and we have used different techniques to rank and select sentences for summary inclusion. Cosine similarity is used to calculate the similarities between sentences and PageRank is used to rank the sentences based on their similarities. We have evaluated our system both subjectively and objectively by using the data we gathered from different sources. The results of both objective and subjective evaluations have shown relatively consistent about the effectiveness of the summarizer. The summarizer scores *83.33%* informativeness, *78.8%* referential integrity and non-redundancy, and 76.66% structure and coherence. Objective evaluation also shows good result, the system gave *0.527* Precision ,*0.422* Recall and *0.468* F-measure when stemmer and other language specific lexicons are combined with word embedding.

However, by using similar data and evaluation method used in the previous work [8] the current summarizer outperformed by *0.052* Precision, *0.064* Recall and 0.058 F-measure.

In our findings we get that, language specific lexicons can improve the overall performance of the summarizer. However, the evaluation was carried out on relatively small data sets and, therefore, this work needs a further development and testing.

7.3 Contribution of This Work

Our contributions are listed below:

- We have developed domain independent, single document text summarization for Afaan Oromo.
- We have developed a pre-trained Word2Vec model which can be used in different domains. This can be used as a starting stone for other broad researches regarding word embedding on Afaan Oromo texts.
- As far as our knowledge our work is the first to try to combine word embedding with PageRank algorithm for Afaan Oromo Text summarization.
- We have conducted different experiments to see the impact of the features used in this work.
- We evaluated the system performance by applying different metrics in accordance with the previous works.

7.4 Future Work

Depending on the findings and knowledge acquired from the study we forward the following recommendations:

- There is no well-prepared and balanced corpus for Afaan Oromo to use for natural language processing study. So, we strongly recommend the preparation of this corpus as it is essential part for further development and evaluation of the performances of the summarizers.
- In addition to that We also recommend complete stop-word list, synonyms, and abbreviations are very useful to enhance term frequency-based method.
- Since there is no research conducted in Afaan Oromo using word embedding we also strongly recommend collecting good Afaan Oromo texts and developing a Word2Vec model which can be used in researches conducted using word embedding.
- This work can be used as a starter and can be improved by adding features and also by changing techniques used. We recommend using neural networks to develop summarizer for the future.

References

[1] Y. Kang, A. Kamal, X. Yanmin, and Z. Zuping, "An Integrated Graph Model for Document Summarization," *School of Information Science and Engineering, Central South University, Changsha 410083, China,* 2018.

[2] P. Anttila, "Automatic Text Summarization," pp. 7-8, May, 2018.

[3] O. Shiyan, S. G. Christopher, K. Dion, and H. Goh, "Automatic Multi-document Summarization of Research Abstracts: Design and User Evaluation," *Journal of the American Society for Information Science & Technology,* 2007.

[4] J. Brownlee, "Machine Learning Majesty," 29 November 2017. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-text-summarization. [Accessed 26 October 2018].

[5] D. Karani, "Towards Data Science," 8 September 2017. [Online]. Available: https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa . [Accessed 25 October 2018].

[6] J. Brownlee, "Machine Learning Mastery," 6 October 2017. [Online]. Available: https://machinelearningmastery.com/develop-word-embeddings-python-gensim. [Accessed 25 October 2018].

[7] Wikipedia, "Wikipedia," 13 October 2018. [Online]. Available: http://www.oromo-people-Wikipedia.com. [Accessed 27 October 2018].

[8] Fiseha Berhanu, "Afaan Oromo News Text Summarizer Based on Sentence Selection Function," *Unpublished Master's Thesis, Addis Ababa University, Ethiopia,* 2013.

[9] H. P. Luhn, "The automatic creation of literature abstracts," *IBM Journal of Research and Development, V*ol. 2, pp. 159–165, 1958.

[10] H. Edmundson, "New Techniquesin Automatic Extracting," *Computing, V*ol. 16, No. 2, pp. 264-285, 1969.

[11] A. Padmakumar and A. Saran, "Unsupervised Text Summarization Using Sentence Embeddings".

[12] A. W. Samer, A. K. Naseer, C. Bolin, and S. Xuequn, "Multidocument Arabic Text Summarization Based on Clustering and Word2Vec to Reduce Redundancy," *MDPI Journal Information, V*ol. 11, No. 59, 2020.

[13] R. Gaetano , B. Pierpaolo, and S. Giovanni , "Centroid-based Text Summarization through Compositionality of Word Embeddings," *Proceedings of the MultiLing 2017 Workshop on Summarization and Summary Evaluation Across Source Types and Genres,* pp. 12–21, 2017.

[14] K. Chandra, S. Gyanit, and P. Nish, "Abstractive and Extractive Text Summarization using Document Context Vector and Recurrent Neural Networks," *KDD 18 Deep Learning Day, V*ol. 2, 2018.

[15] Kamil Nuru, "Automatic Amharic NewsText Summarizer," *Unpublished Master's Thesis, Addis Ababa University, Ethiopia,* 2005.

[16] Eyob Delele, "Topic-based Amharic Text summarization," *Unpublished Master's Thesis, Addis Ababa University,* 2011.

[17] Kifle Deresse, "Graph-based Automatic Amharic Text Summarizer," *International journal of Scientific and Engineering research, V*ol. 8, 2017.

[18] Melese Tamiru, "Automatic Amharic Text Summarization Using Latent Semantic Analysis," *Unpublished Master's Thesis, Addis Ababa University, Ethiopia,* 2009.

[19] Girma Debele, "Afaan Oromo news text summarizer," *Unpublished Master's thesis, Addis Ababa University, Ethiopia,* 2012.

[20] Asefa Bayisa, "Query-based Automatic Summarizer for Afaan Oromo Text," *Unpublished Master's thesis, Addis Ababa University, Ethiopia,* 2015.

[21] K. Anita R., "An Efficient Domain-Specific Text Summarization Technique using Knowledge-Base and Combined Statistical and Linguistic Methods," *Unpublished Doctoral Thesis,Solapur University, Solapur,* 2016.

[22] S. Jagadish, "Summarizing News Paper Articles: Experiments with OntologyBased,," *Cybernetics and Information Technologies, V*ol. 12, No. 2, 2012.

[23] J.-M. Torres-Moreno, Automatic Text Summarization, London: ISTE Ltd and John Wiley & Sons, Inc, 2014.

[24] A. Archana and C. Sunitha, "An Overview on Document Summarization Techniques," *International Journal on Advanced Computer Theory and Engineering (IJACTE), V*ol. 1, No. 2, pp. 2319 – 2526, 2013.

[25] E. Lloret, "Text summarization: An overview," *Spanish Government under the project TEXT-MESS,* 2006.

[26] J. Romero, "Abstractive Text Summarisation with Neural Networks," *UnPublished Masters Thesis, Data Analytics Lab, ETH Zürich,* 2017.

[27] H. Lucas de, "Extractive Summarization using Sentence Embeddings," *Unpublished Masters Thesis, Department of Information and Computing Sciences, Fucalty of Science, Utrecht University,* 2017.

[28] D. Sarkar, "Towards Data Science," A Medium publication sharing concepts, ideas, and codes., 14 04 2018. [Online]. Available: https://towardsdatascience.com/understanding-feature-engineering-part-4-deep-learning-methods-for-text-data-96c44370bbfa. [Accessed 12 03 2020].

[29] D. Sarkar, "Traditional Methods for Text Data," Towards Data Science, 30 01 2018. [Online]. Available: https://towardsdatascience.com/understanding-feature-engineering-part-3-traditional-methods-for-text-data-f6f7d70acd41. [Accessed 02 12 2019].

[30] J. K. Raulji, and R. S. P. Jatinderkumar, "Stop-Word Removal Algorithm and its Implementation for Sanskrit Language," *International Journal of Computer Applications, V*ol. 150, No. 2, 2016.

[31] C. Mallick, A. K. Das, M. Dutta, and A. Sarkar, "Graph-Based Text Summarization Using Modified TextRank," *Soft Computing in Data Analytics. Advances in Intelligent Systems and Computing, V*ol. 758, 2019.

[32] H. Mujtaba, "An Introduction to Bag of Words (BoW) | What is Bag of Words?," greatLearningblog, 20 04 2020. [Online]. Available: https://www.mygreatlearning.com/blog/bag-of-words. [Accessed 14 04 2020].

[33] E. Gunes and R. Dragomir, "LexRank: Graph-based Lexical Centrality as Salience in Text Summarization," *Journal of Artificial Intelligence Research, V*ol. 22, pp. 457-479, 2004.

[34] Marco Baroni, Georgiana Dinu, and Germán Kruszewski, "Don't count, predict! A systematic comparison of context-counting vs. context-predicting semantic vectors," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, V*ol. 1, pp. 06, 2014.

[35] T. Mikolov, C. Greg, J. Dean, I. Sutskever, and K. Chen, "Efficient Estimation of Word Representations in Vector Space," *Advances in Neural Information Processing Systems, V*ol. 3, 2013.

[36] T. Mikolov, C. Greg, J. Dean, I. Sutskever and K. Chen, "Distributed Representations of Words and Phrases and their Compositionality," *Advances in Neural Information Processing Systems, V*ol. 1, 2013.

[37] J. Brownlee, "How to Use Word Embedding Layers for Deep Learning with Keras," 03 10 2019. [Online]. Available: https://machinelearningmastery.com/use-word-embedding-layers-deep-learning-keras.

[38] D. Radev, H. Jing, M. Stys, and D. Tam, "Centroid-based Summarization of Multiple Documents," *Information Processing & Management, V*ol. 40, No. 6, pp. 919-938, 2004.

[39] F. Yimai and H. Hughes , "Proposition-based summarization with a coherence-driven incremental model," *Doctoral Dissertation, University of Cambridge,* 2018.

[40] T. Zhi, L. Ye, R. Fuji, and T. Seiji, "Single Document Summarization Based on Local Topic Identification and Word Frequency," *Seventh Mexican International Conference on Artificial Intelligence, V*ol. 7, No. 08, 2008.

[41] L. Page and S. Brin, "The Anatomy of a Large Scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems, V*ol. 30, pp. 107- 117, 1998.

[42] J. Brownlee, "Encoder-Decoder Models for Text Summarization in Keras," Machine Learning Mastery, 07 08 2019. [Online]. Available: https://machinelearningmastery.com/encoder-decoder-models-text-summarization-keras/. [Accessed 12 03 2020].

[43] . C. Jianpeng and L. Mirella, "Neural Summarization by Extracting Sentences and Words," *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, V*ol. 1, pp. 484–494, 2016.

[44] S. Josef and J. Karel, "Evaluation Measures for Text Summarization," *Computing and Informatics, V*ol. 28, No. 2, pp. 1001–1026, 2009.

[45] Tilahun G., "Qubee Afan Oromo : Reasons for choosing the Latin script for developing an Afan Oromo Alphabet," *Journal of Oromo studies,* 1993.

[46] C. Griefenow-Mewis, W. J.G, Möhlig and B. Heine, "A Grammatical Sketch of Written Oromo," *Grammatical Analyses of African Languages,* vol. 16, 2001.

[47] Debela Tesfaye, "Designing a Stemmer for Afan Oromo Text: A hybrid approach",," *Unpublished Master¨s thesis, Addis Ababa University, Ethiopia,* 2010.

[48] Gumii Qormaata Afan Oromoo, Caasluga Afan Oromo, Finfinnee: Komishinii Aadaaf Turizmii Oromiyaa, 1995.

[49] Sanchit Agarwal, Nikhil Kumar Singh and Priyanka Meel, "Single-Document Summarization Using Sentence Embeddings and K-Means Clustering," *International Conference on Advances in Computing, Communication Control and Networking,* 2018.

Appendixes

Appendix A: List of Afaan Oromo Abbreviations

k.k.f	Kan kana fakkaatan	Onk.	Onkololeessa
Obb.	Obboo	Bit.	Biteetossa
Add.	Addee	Mud.	Muddee
Bil.	Biliyoona	Wax.	Waxabajjii
fkn.	Fakkeenyaaf	Gur.	Guraandhala
hub.	Hubaachiisaa	Hag.	Hagayya
w.k.f	Waan kana fakkaatan	Ebl.	Ebla
mil.	Miliyoona	W.B.	Waree booda
ful.	Fulbaana	Ado.	Adoolleessa
Sad.	Sadaasa	W.D.	Waaree dura
Mr.	Mister (in some cases)		
Ama.	Amajjii		

Appendix B: List of Afaan Oromo Stop Words

Aanee	**eega**	**hoggaa**	**Iseef**	**Jala**	**Karaa**	**naa**	**otumallee**	**ta'es**	**yommii**
Adda	**eegas**	**hoggas**	**Iseen**	**Jara**	**ka'uun**	**naaf**	**otuu**	**tahullee**	**yommuu**
Agarsiisoo	**eegana**	**hogguu**	**iseenis**	**jechaan**	**Kee**	**naan**	**otuullee**	**tana**	**yoo**
Akka	**eegasii**	**hogguus**	**Ishee**	**jechoota**	**Keen**	**naannoo**	**qaba**	**tanaaf**	**yookaan**
Akkam	**egasii**	**Hoo**	**Isheef**	**jechuu**	**Kees**	**narraa**	**qabdi**	**tanaafi**	**yookiin**
Akkas	**egasiis**	**iddoo**	**isheen**	**jechuuf**	**Keenna**	**natti**	**qabna**	**tanaafis**	**yookinimoo**
Akkasumas	**ennaa**	**illee**	**Ishii**	**jechuun**	**Keenya**	**ni**	**qabu**	**tanaafuu**	**yoolinimoo**
Akum	**erga**	**immoo**	**Ishiif**	**jechuunis**	**Keenyaa**	**nu**	**saaniif**	**ta'ullee**	**yoom**
Akkuma	**ergasii**	**Ini**	**Ishiin**	**jedha**	**Keessa**	**nus**	**sadii**	**ta'uu**	
Ala	**ergii**	**innaa**	**ishiirraa**	**jedhan**	**Keessan**	**nu'i**	**sana**	**ta'uun**	
Alatti	**F**	**innasuu**	**ishiitti**	**jedhe**	**Keessatti**	**nu'is**	**saniif**	**ta'uyyu**	
Alla	**faallaa**	**inni**	**Isii**	**jedhu**	**Keeti**	**nurraa**	**sanis**	**ta'uyyuu**	
Amma	**fagaatee**	**innis**	**Isiin**	**jette**	**Keetii**	**nurraas**	**si**	**tawullee**	
Ammo	**Fi**	**Irra**	**Isin**	**jetti**	**Keetiis**	**nuti**	**sii**	**teenya**	
Ammo	**fullee**	**irraa**	**Isini**	**jira**	**Keetis**	**nutis**	**siif**	**teessan**	
An	**fuullee**	**irraan**	**Isinii**	**jirtutti**	**Kiyya**	**nutti**	**siin**	**tiyya**	
Ana	**gajjallaa**	**irratti**	**Isiniif**	**jiru**	**Kiyyas**	**nuu**	**silaa**	**tokko**	
Ani	**gama**	**Isa**	**isiniifillee**	**jirutti**	**Koo**	**nuuf**	**silas**	**too**	
Anis	**gara**	**Isaa**	**isiniifis**	**ka**	**Koof**	**nuun**	**simmoo**	**tti**	
Ati	**gararraa**	**isaaf**	**Isiniin**	**kaa'uun**	**Koos**	**nuy**	**sinitti**	**ture**	
Bira	**garas**	**isaan**	**isiniinis**	**kan**	**Kun**	**nuyi**	**siqee**	**utuu**	
Booda	**garuu**	**isaaniif**	**isinillee**	**kana**	**Kunis**	**nuyis**	**sirraa**	**utuullee**	
Boodas	**giddu**	**isaanis**	**isinirraa**	**kanaa**	**Kuniyyuu**	**odoo**	**sitti**	**waa'ee**	
Booddee	**gidduu**	**isaani**	**isinirraas**	**kanaaf**	**Lafa**	**ofii**	**sun**	**waan**	
Booddees	**gubbaa**	**isaanii**	**Isinis**	**kanaafi**	**Lama**	**ofiis**	**sunis**	**waggaa**	
Dabalatees	**ha**	**isaaniitiin**	**isinitti**	**kanaafillee**	**Malee**	**oggaa**	**sunniin**	**wajjin**	
Dhaan	**hamma**	**isaanirraa**	**isinittis**	**kanaafis**	**Malees**	**oggas**	**ta'a**	**waliin**	
Dudduuba	**hanga**	**isaanitti**	**ittaanee**	**kanaafiyyuu**	**Manna**	**oo**	**ta'aa**	**warra**	
Dugda	**hangas**	**isaatiin**	**Itti**	**kanaafuu**	**Maqaa**	**ol**	**ta'an**	**woo**	
dura	**henna**	**isarraa**	**itumallee**	**kanaan**	**Miti**	**osoo**	**ta'e**	**yammuu**	
duras	**hennas**	**isatti**	**Ituu**	**kanaatti**	**Moo**	**otoo**	**ta'ee**	**yemmuu**	
duuba	**hin**	**Isee**	**ituullee**	**kanatti**	**Na**	**otuma**	**ta'eef**	**yeroo**	

Appendix C: List of Afaan Oromo Verb Affixes

a	adhee	amtu	atte	dani	ine	itani	nnu	ta	uuttan
aa	adhuu	amtuu	atti	de	inu	ite	nu	tani	uutti
aaf	adhuu	amu	attu	dha	is	iti	ooftna	taniittu	xa
aas	ama	amuu	atu	dhe	isa	itu	oofta	te	xani
aat	amaa	amuudhaa	chiisa	dhu	isan	ja	ooftan	tetta	xe
aatii	aman	amuudhaf	chiisan	di	ise	jani	oofte	teetii	xi
aatu	amani	amuuf	chiise	du	isisa	je	oofti	ti	xu
achisa	amanii	amuun	chiisna	duu	isise	ju	ra	tu	xuu
achiisan	ame	ani	chiisne	e	isisna	la	re	tuu	
achiise	amne	aniiru	chiiste	eera	isista	le	ru	u	
achisna	amni	anna	chisiisa	ees	isistan	lu	se	ulle	
achistan	amoo	anne	chisiisan	eet	isiste	na	sisa	umsa	
achiste	amta	annu	chisiista	eeti	isna	naan	sisan	uu	
achuu	amtan	ata	chisiistan	i	istan	ne	sise	uuf	
achuuf	amtani	atani	chisiiste	ifna	iste	neerra	sisna	uufan	
adha	amte	ate	chisiistu	ifte	isu	ni	sisne	uufi	
adhe	amti	atini	da	ina	ita	nna	siste	uufii	

Appendix D: Noun Suffixes

aa	eeyyii	iinis	irratti	llee	ooliin	s
aaf	f	iis	irrattillee	n	ooliiwwan	tii
an	I	illee	irrattis	ni	ooma	tiin
aniif	icha	irraa	irrattuu	oolee	oota	tu
aniin	ichi	irraahille	itti	ooleedhaan	ootaaf	uma
arraa	ii	irraahis	ittii	ooleef	ootaan	umaa
atti	iif	irraahuu	ittiin	ooleen	ootadhaan	umaaf
dhaa	iifis	irraan	ittillee	ooleewwan	ootawwan	umaafillee
dhaaf	iifuu	iraannille	ittis	oolii	ootni	umaanille
dhaan	iin	iraanis	ittuu	ooliidhan	oottan	umaanis
een	inille	irraanuu	lee	ooliif	rra	umaanuu

Appendix E: **Validation Summary Preparation Guideline**

Addis Ababa University

College of Natural Science

Department of Computer Science

Dear respondent,

The purpose of this questioner is to design Automatic Afaan Oromo text summarizer. The system generates extractive type of summary for each of the input text. An extractive summary is created by selecting a certain number of sentences that are judged to be the most important out of the original text.

Hence, this appendix describes guideline and instructions that you follow to prepare summary of a given topic. You are requested to form an extractive summary for each of the text you are given. When sentences are selected for inclusion in the summary all that needs to be considered is the importance of the sentences. Furthermore, you can rank the sentences for inclusion in a summary. The summary should not be more the 40% of the original text. If the original text has 15 sentences the summary should not be more than 6 sentences (15*0.4=6).

Dear respondent! You are expected to read the original news items carefully until you understand the concepts. Then you are going to rank the sentences according to their importance.

Thank you for your help!

Topic 1

Aartiistoonni Oroomoo akka saba warra biraa magaalaa Finfinnee bakkeewwan gurguddaatti akka hojiisaanii hindhiyeesiineef dandeettiin maallaqaa nuu daanggeesseera jechuun BBC'tti himte Artiisti Hawwii Tazarraa. Aartiisitiin Oromoo beekamtuu Hawwiin, baandii muziiqaa isheetiin alattis sagantaa 'Badhaasa Siiqee' jedhuu eegaluudhaan dubartoota Oromoo karaa hedduudhaan seenaa Oromoo keessatti aarsaa kanfalaniif beekamtii kennuu eegaltee jirti.

Maaliif akka maqaa sagantaa kana akka filattee yoo himtuus, "Siiqeen aadaa Oromoo keessatti mallattoo nageenyaafi kan eddoo olaanaa qabuudha. Dubartiin Siiqee qabattee baanaan waraana illeen qabanneessu dandeessi." jetti. Kanaafuu, yaadni Badhaasa Siiqee kunis kan baroota dheeraaf na keessa tureedha kan jettu aartiisti Hawwiin, wallistoonni, ateeletoonni fi kanneen ogummaa Oromoo garaa garaa keessatti hirmaachuun artiifi aadaa akkasumas ummataaf waan guddaa hojjatanii fi aarsaa olaanaa kanfalan jiru jetti. "Kanaaf jecha anis maqaa kanan moggaafadhee jetti" artisit Hawwii Tazarraa.

Ka'umsa badhaasa Siiqee yoo ibsituus, yaaduma kanaaf jecha baatii sadiin dura Baandii Hawwii Tazarraa jedhuun hayyama baafachuun, badhaafamtoota damee xabatoota meeshaa shamarranii fi weellistoota shamarranii fi ateeleetoota kan hammateedha jetti. Haa ta'u malee, akka warraa kaanii eenyuu yaa mo'atu jette adda baasuuf carraa dhabdulleen wellistootaa fi kanneen guddina aadaa fi aartii Oromootiif aarsaa olaanaa kanfalan galateenfachuudha kaayyoo guddaan badhaasa kanaa. "Jireenyatti wal galateefachuu qabna. Oromoon sabaaf waan guddaa gumaatee tokko osoo lubbuun jiruu galateenfamu qaba jedheen sagantaa kana qopheessuu eegale," jetti Haawwiin.

Haaluma kanaanis sirna badhaasaa Muddee 30 bara 2018 Magaalaa Finfinnee Galma Aadaa Oromootti gaggeefameerratti atileetoota dubartoota Oromoo kana dura qabxii olaanaa galmeessuu maqaa biyyattii waamsiisa garuu amma eessa buuteen isaanii ummanni quba hinqabne, artiistoota dubartoota Oromoo, Doktoroota dubartii Oromoo fa'ii kan hammate ta'u himti. Haaluma kanaanis ateeletoota shaniif beekamtiin kennameera. Isaanis Daraartuu Tulluu, Faaxumaa Roobaa, Geexee Waamii, Quxuree Dullachaa fi Biraanee Adaree waan Oromoo mataa ol qabachiisaniif badhaasni kun kennameefiira jetti, Hawwiin.

Akka Hawwiin jettuuttis, shamarreen Oromoo kunneen isaan aarsaa olaanaa kanfaluun seenaa hojatanii akka ofiitti boonnu nu taasisaniidha. Isaanis lubbuun jiraatanii kana arguun isaanii gammachu hunda caaluudhas jette jirtu. Akkasumas Artisti Ilfinash Qannoo, Haalloo daawwii,

fi Angaatuu Baalchaa gootoota dubartoota akka of barruuf aarsaa olaanaa taasisan yoo ta'an, Ababach Daraaraa amma kan lubbuun hinjirre ta'an illeen gumaati isaan taasisan kan bira hindabarreedha ture jetti Hawwiin. Artisti Yashii Raggaasaa fi Ababachi Ajjamaa kanneen aartiidhaan bara dheeraaf aadaa Oromoo guddiisuu keessatti shoora olaanaa xabataniitu badhaasni kun keennameef, jeetti Hawwiin.

Artiistoonni Oromoo miseensa jarmiyaa kamiiyyu miti kan jettu Hawwiin, kan isaan lalaban ammmoo waa'ee miidhaa ummata Oromoo ta'u himti. Kanaafu, hawaasni Oromoo aartistoota bira dhaabbachuun isaan deegaruun guddina aartiifi dagaaginsa aadaa Oroomoo akkasumas fiixaan bahiinsa hawwii isaanii ni saffisa jetti. Badhaasni Siiqee tokkooffan bara kana kaka'umsa dhuunfaa Hawwiitii eegalamee kunis, gootoota dubartoota Oromoo seektara kammiiyu irratti argaman garuu kanneen guddinaafi dagaagna aadaafi artiifi gumaatan akkasumas Oromoof waan gaarii hojjatan hunda hammatas jetti.

Dubartootn badhaasichaaf filataman kanneen waajjiraalee motummaa keessa hojjatan illeen ta'u mala kan jette Hawwiin, badhaasa kanas Finfinnee qofatti osoo erga baandii isheetiin Oromiyaarra naannooftee beeksiifteen booda, biyyoota ollaa biraatti illeen raawwachuuf karoora akka qabdu himtii. Aartiiftoonni Oromoo akka aartiiftoota saba birootti maagaalaa Finfinnee keessatti hojjachuuf dandeettiin maallaqaa isaanii hinheemuuf kan jettu hawwiin, haalli kunis akka hojiiwwan aartii isaanii bakkeewwan gurguddaatti akka hinhojjaneef daangaa itti umuusaa himti. "Anillee galma barkumeetti hojjachuuf yaaleen ture, garuu hinmilkoofne. Garuu Waaqni jedhee wantoota abdii nama kutachiisan hedduu keessatti milkaa'eera" jetti.

Appendix F: **System summary evaluation guide line**

Addis Ababa University
College of Natural Science
Department of Computer Science

Dear respondent,

The purpose of this questioner is to evaluate the performance of Automatic Afaan Oromo news text summarizer. The system generates extractive type of summary for each of the input text. An extractive summary is created by selecting a certain number of sentences that are judged to be the most important out of the original text.

Three different summaries are generated at the end of the topic after you read the summary evaluate the summary based on the three-question listed below. Fill the number given for the summary.

For example, if the summary informativeness of summary 1 is good check(X) the box provided under choice Good.

1. **The summary informativeness:**

Very good	Good	Not Bad	Poor	Very Poor
☐	☐	☐	☐	☐

2. **Grammar, non-redundancy and referential clarity.**

Very good	Good	Not Bad	Poor	Very poor
☐	☐	☐	☐	☐

3. **Structure and Coherence**

Very good	Good	Not Bad	Poor	Very Poor
☐	☐	☐	☐	☐

Note:

- **Informativeness**: The best sentences are that contain the most important information of the topic sentence
- **Non-redundancy: -** The summary should not contain redundant information.
- **Referential clarity: -** The nouns and pronouns should be referred to in the summary. For example, the pronoun ‘she’ has to mean that it is referring to somebody in the context of the summary.
- **Structure and Coherence: -** The summary should have good structure and the sentences should be coherent.

Thank you for your help!

Topic2

Hojii egereen fiilmii Afaan Oromoo abdachiisaa ta'uu akeeke.

Fiilmiin 'Miixuu' jedhu Dilbata darbe Finfinnee galma Aadaa Oromootti eebbifame egereen industirii fiilmii Afaan Oromoo abdachiisaa ta'uu agarsiiseera jedhan ogeessonniifi hirmaattonni fiilmichaa. Barreessaafi daarekterri fiilmichaa dargaggoo Biqilaa Asfaaw bu'aa ba'ii 'Miixuu' hojjechuuf keessa darbe BBCf yeroo ibsetti taatota hanga ammaatti hin kaffaliiniif jiruuf hojii ofii isaa hojjete osoo hin shallagiin qarshii 300,000 akka itti baase dubbata.

Fiilmiin sa'aa 1:50 dheeratuu kuni taatee bara Dargii keessa rawwatame ka'umsa godhachuun seenaa maatii tokkoo hima. Yaaddoon haadha ilmoon jalaa baddeefi qorumsi jireenyaas dhimmoota fiilmi Miixuu keessatti ka'an keessaa tokkodha.

Rakkoo hojii isaa keessatti isa mudate ijoonis meeshaa guutuu kan akka istuudiyoo pirodaakshiniifi dhaabbileen fiilmii Afaan Oromoo ispoonsara gochuuf amanta dhabuu akka ta'e dubbata Biqilaan. Rakkoo jiran kana mo'achuun ogeessota jiru jedhamaniifi taatota buleeyyii akkasumas isoonsara dhaabbileen guddinaafi dagaagina fiilmii Afaan oromoof aantummaa qaban hirmaachisee milkeesseera Biqilaan.

"Fiilmichi jaalalaafi siyaasa dhokataa of keessaa qaba," kan jedhu barreessaafi qopheessaan fiilmichaa dargaggoo Biqilaa, naannoo Amaaraatti magaala Kamisee dabalatee godinaalee Oromiyaa garaagara keessatti agarsiisuuf karoorfachuusaa BBC'tti himeera. Soolan Adimaasuu fiilmii 'Miixuu' kana keessatti hiiroodha ykn qooddataa cimaadha. Diraamaa dheeraa 'Dheebuu' keessattis maqaa 'Guddaa' jedhamuun kan beekamu Soolan, waa'ee fiilmii kanaa BBC'tti himeera.

Fiilmiin 'Miixuu' fiilmiiwwan Afaan Oromoo hanga ammaatti hojjetaman keessaa waan baayyeen adda isaa taasisa kan jedhu Soolan akkas ta'e jedhee akka hin yaadne dubbata. Akka inni jedhutti, seenaa fiilmii kanaa baayyee gaarii kan ta'eefi kaameraa sadarkaa isaa eeggateen waan hojjetameef qulqullina gaarii qaba. Daarikterri fiilmii kanaa Biqilaan hojii gaarii akka hojjete kan himu Soolan, 'Miixuun' seenaa isaarraa kaasee, akka inni itti ijaaramee, bilchinni ittiin hojjetameefi akkaataa ittiin daayirekti ta'ellee adda isa taasisas jedheera.

Bakki filatamee itti hojjetames horii qusachuudhaaf bakuma argameti kan waraabame osoo hin taane bakkuma sirriitti waan ta'eef hedduu miidhagaadha jedha. Kan biraan kan 'Miiixuu' adda taasisu, ispponsera kanaan dura hin baramne sadan arfan tokko qaba jedha Soolan. "Kun

ammoo akka mucaan kun dabalee hojjetuuf kaka'umsa ta'aaf jedheen yaada." Shaakala irraa kaasee hedduu itti dadhabamuu kan dubbatu Soolan, "akkaataa taatonni itti filataman, waanti hundumti sagaleefi qulqullina pirodaakshinii dabalatee fiilmii Afaan Oromoo kanaan dura hojjetamerraa adda isa taasisa," jedha.

Soolan, namoonni eebba fiilmichaa irratti argaman hanga yaadameen ol hedduu akka ta'e yaadatee, yaadni namoonni kennaniifis kan isaan gammachiise ture jedha. "Galmi nuti qabnu lachuu guutee kaan dhaabbatanii daawwachaa turan kaan ammoo bakkallee dhabanii deebiyan galan. Kun ammoo waan nama gammachiisudha." Fiilmiin 'Miixuu' namni dhimmee hojjechuu danda'u akka sadarkaa addunyaatti dorgomuu danda'u abdii guddaa namatti agarsiisa jedha Soolan.

Yoomis Gonfaa ammoo Ediitera filmii 'Miixuu'dha. Yoomisillee diraamaa Afaan Oromoo dheeraa 'Hiree' jedhu OBN irratti darbaa jiruu dabalatee diraamaawwan gaggabaaboo akkasumas fiilmii gara garaa gulaaleera. Fiilmii 'Miixuu' dhugoomsuudhaaf hedduu akka dhimman dubbatee, "Yaada namoonni eebba fiilmichaarratti kennaa turan firii dadhabii keenyaatti akka gamadnu nu taasisedha," jedha Yoomis.

Namoonni fiilmicha ilaalan hedduun "Fiilmii sadarkaa fiilmiin Afaan Oromoo irra jiru gulantaa tokko ol guddise" jechuun dinqisiifannaa isaanii marsaalee hawaasaa irratti ibsataa jiru. "Fiilmiin MIIXUU fiilmii baay'ee bareedaafi waan baay'ee irraa barreedha" jechuun fuula 'Facebook' isaarratti kan barreessee ogeessi fiilmii Qalbeessaa Magarsaa, ogeessonni ogummaa kanatti bobba'an waan bareeda irraa baratu jedhaan yaada jedheera.

Walumaa galatti fiilmiin kun si'oomina artii Oromoo akka dabalu kan taasise ta'uu dubbatanii, ogeessonni gita gitaan jiran kaan ifaa ifarratti, kan barreessu barreessuu irratti, kan sagalee sagalee irratti xiyyeeffatanii yoo hojjetaa deeman fiilmiin Afaan Oromoo kana caalaa foyyaa'aa deema jedhan. Dargaggoo Biqilaa Asafawu diraamaa raadiyoo 'Imimmaan jaalalaa' fi kan televiziyoonaa 'Har'i boru miti' jedhan barreessun beekkama.

Fiilmiin 'MIIXUU' namoonni Afaan Oromoo hin beekne akka daawwataniif jecha afaan Ingiliffaan jalatti hiiknis kennamuu himeera darggaggoo Biqilaa. Fiilmii kana keessatti artiistotni uleeyyiifi haaraan kan hirmaatan yommuu ta'u, artistota buleeyyi keessaa Abbabach Ajjamaa, Humneechaa Asaffaa, Zinnaash Olaaniifi Soolan Admaasuu ni argamu.

Summary 1

Akka inni jedhutti, seenaa fiilmii kanaa baayyee gaarii kan ta'eefi kaameraa sadarkaa isaa eeggateen waan hojjetameef qulqullina gaarii qaba. Kun ammoo akka mucaan kun dabalee hojjetuuf kaka'umsa ta'aaf jedheen yaada. Yoomisillee diraamaa Afaan Oromoo dheeraa 'Hiree' jedhu OBN irratti darbaa jiruu dabalatee diraamaawwan gaggabaaboo akkasumas fiilmii gara garaa gulaaleera. Fiilmiin MIIXUU fiilmii baay'ee bareedaafi waan baay'ee irraa barreedha" jechuun fuula 'Facebook' isaarratti kan barreessee ogeessi fiilmii Qalbeessaa Magarsaa, ogeessonni ogummaa kanatti bobba'an waan bareeda irraa baratu jedhaan yaada jedheera. Walumaa galatti fiilmiin kun si'oomina artii Oromoo akka dabalu kan taasise ta'uu dubbatanii, ogeessonni gita gitaan jiran kaan ifaa ifarratti, kan barreessu barreessuu irratti, kan sagalee sagalee irratti xiyyeeffatanii yoo hojjetaa deeman fiilmiin Afaan Oromoo kana caalaa foyyaa'aa deema jedhan.

Summary 2

Rakkoo jiran kana mo'achuun ogeessota jiru jedhamaniifi taatota buleeyyii akkasumas isoonsara dhaabbileen guddinaafi dagaagina fiilmii Afaan oromoof aantummaa qaban hirmaachisee milkeesseera Biqilaan. Soolan Adimaasuu fiilmii 'Miixuu' kana keessatti hiiroodha ykn qooddataa cimaadha. Akka inni jedhutti, seenaa fiilmii kanaa baayyee gaarii kan ta'eefi kaameraa sadarkaa isaa eeggateen waan hojjetameef qulqullina gaarii qaba. Kun ammoo akka mucaan kun dabalee hojjetuuf kaka'umsa ta'aaf jedheen yaada. Yoomisillee diraamaa Afaan Oromoo dheeraa 'Hiree' jedhu OBN irratti darbaa jiruu dabalatee diraamaawwan gaggabaaboo akkasumas fiilmii gara garaa gulaaleera. Fiilmiin MIIXUU fiilmii baay'ee bareedaafi waan baay'ee irraa barreedha" jechuun fuula 'Facebook' isaarratti kan barreessee ogeessi fiilmii Qalbeessaa Magarsaa, ogeessonni ogummaa kanatti bobba'an waan bareeda irraa baratu jedhaan yaada jedheera. Walumaa galatti fiilmiin kun si'oomina artii Oromoo akka dabalu kan taasise ta'uu dubbatanii, ogeessonni gita gitaan jiran kaan ifaa ifarratti, kan barreessu barreessuu irratti, kan sagalee sagalee irratti xiyyeeffatanii yoo hojjetaa deeman fiilmiin Afaan Oromoo kana caalaa foyyaa'aa deema jedhan.

Summary 3

Rakkoo jiran kana mo'achuun ogeessota jiru jedhamaniifi taatota buleeyyii akkasumas isoonsara dhaabbileen guddinaafi dagaagina fiilmii Afaan oromoof aantummaa qaban

hirmaachisee milkeesseera Biqilaan. Soolan Adimaasuu fiilmii 'Miixuu' kana keessatti hiiroodha ykn qooddataa cimaadha. Akka inni jedhutti, seenaa fiilmii kanaa baayyee gaarii kan ta'eefi kaameraa sadarkaa isaa eeggateen waan hojjetameef qulqullina gaarii qaba.Daarikterri fiilmii kanaa Biqilaan hojii gaarii akka hojjete kan himu Soolan, 'Miixuun' seenaa isaarraa kaasee, akka inni itti ijaaramee, bilchinni ittiin hojjetameefi akkaataa ittiin daayirekti ta'ellee adda isa taasisas jedheera. Bakki filatamee itti hojjetames horii qusachuudhaaf bakuma argameti kan waraabame osoo hin taane bakkuma sirriitti waan ta'eef hedduu miidhagaadha jedha. Kun ammoo akka mucaan kun dabalee hojjetuuf kaka'umsa ta'aaf jedheen yaada. Fiilmiin 'Miixuu' namni dhimmee hojjechuu danda'u akka sadarkaa addunyaatti dorgomuu danda'u abdii guddaa namatti agarsiisa jedha Soolan. Yoomis Gonfaa ammoo Ediitera filmii 'Miixuu'dha. Yoomisillee diraamaa Afaan Oromoo dheeraa 'Hiree' jedhu OBN irratti darbaa jiruu dabalatee diraamaawwan gaggabaaboo akkasumas fiilmii gara garaa gulaaleera. Fiilmiin MIIXUU fiilmii baay'ee bareedaafi waan baay'ee irraa barreedha" jechuun fuula 'Facebook' isaarratti kan barreessee ogeessi fiilmii Qalbeessaa Magarsaa, ogeessonni ogummaa kanatti bobba'an waan bareeda irraa baratu jedhaan yaada jedheera. Walumaa galatti fiilmiin kun si'oomina artii Oromoo akka dabalu kan taasise ta'uu dubbatanii, ogeessonni gita gitaan jiran kaan ifaa ifarratti, kan barreessu barreessuu irratti, kan sagalee sagalee irratti xiyyeeffatanii yoo hojjetaa deeman fiilmiin Afaan Oromoo kana caalaa foyyaa'aa deema jedhan.

Appendix G: Subjective test data evaluation result

1. Summary information evaluation result

Test/DocId	Result			
	S1	S2	S3	S4
Test 2	4	5	5	4
Test 4	5	3	4	4
Test 9	4	5	5	4
Test 12	5	5	4	4
Test 13	3	3	3	3
Test 18	5	4	5	4
Test 19	5	5	5	5
Test 21	4	3	3	4
Test 22	3	5	4	4

2. Referential integrity and Non redundancy summary result

Test/DocId	Result			
	S1	S2	S3	S4
Test 2	4	5	5	4
Test 4	4	3	4	3
Test 9	4	4	4	4
Test 12	4	5	5	4
Test 13	4	3	3	4
Test 18	5	5	5	5
Test 19	5	5	4	4

Test 21	3	3	3	3
Test 22	3	3	3	3

3. Coherence Summary result

Test/DocId	**Result**			
	S1	**S2**	**S3**	**S4**
Test 2	4	4	4	4
Test 4	4	3	4	3
Test 9	4	3	3	4
Test 12	4	4	5	3
Test 13	4	3	3	2
Test 18	5	5	4	4
Test 19	5	5	4	4
Test 21	5	3	4	4
Test 22	4	3	3	4

Declaration

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

Declared by:

Name: ***Lamesa Tashoma Fanache***

Signature: _____

Date: _____

Confirmed by advisor:

Name: ***Yaregal Asabie (PhD)***

Signature: _____

Date: _____